



VYSOKÁ ŠKOLA BÁŇSKÁ – TECHNICKÁ UNIVERZITA OSTRAVA  
EKONOMICKÁ FAKULTA

KATEDRA APLIKOVANÉ INFORMATIKY

Analýza, návrh a implementace databázového systému pro logistickou firmu

Analysis, Design and Implementation of the Database System for Logistic  
Company

Student:  
Vedoucí bakalářské práce:

Tereza Pavlíková  
Ing. Vítězslav Novák, Ph.D.

Ostrava 2013

VŠB - Technická univerzita Ostrava  
Ekonomická fakulta  
Katedra aplikované informatiky

## Zadání bakalářské práce

Student: **Tereza Pavlíková**

Studijní program: B6209 Systémové inženýrství a informatika

Studijní obor: 6209R001 Aplikovaná informatika

Téma: Analýza, návrh a implementace databázového systému pro logistickou firmu  
Analysis, Design and Implementation of the Database System for Logistic Company

Zásady pro vypracování:

1. Úvod
  2. Teoretická východiska návrhu databázových systémů
  3. Analýza současného stavu
  4. Návrh a implementace databázového systému
  5. Závěr
- Seznam použité literatury  
Seznam zkratk  
Prohlášení o využití výsledků bakalářské práce  
Seznam příloh  
Přílohy

Seznam doporučené odborné literatury:

CONOLLY, T., C. E. BEGG a R. HOLOWCZAK. *Mistrovství - databáze: profesionální průvodce tvorbou efektivních databází*. Brno: Computer Press, 2009. ISBN 978-80-251-2328-7.

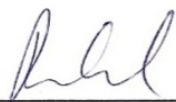
HERNANDEZ, Michael J. *Návrh databází: základy databází : krok za krokem*. Praha: Grada, 2006. ISBN 80-247-0900-7.

HOTEK, M., C. E. BEGG a R. HOLOWCZAK. *Microsoft SQL Server 2008: krok za krokem*. Brno: Computer Press, 2009. ISBN 978-80-251-2466-6.

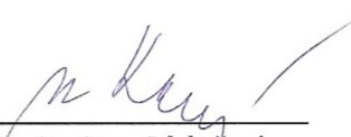
Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Vítězslav Novák, Ph.D.**

Datum zadání: 23.11.2012  
Datum odevzdání: 10.05.2013

  
Ing. Petr Rozehnal, Ph.D.  
vedoucí katedry



  
prof. Dr. Ing. Dana Dluhošová  
děkanka fakulty

Prohlašuji, že jsem celou práci, včetně příloh, vypracovala samostatně.

V Ostravě dne 10. 5. 2013

*Patricia*

.....

Na tomto místě bych chtěla poděkovat svému vedoucímu bakalářské práce,  
Ing. Vítězslavu Novákovi, Ph.D., za odborné vedení, cenné rady a připomínky, které mi  
pomohly k vypracování této bakalářské práce.

# Obsah

1	Úvod .....	5
2	Teoretická východiska návrhu databázových systémů .....	7
2.2	Databázový systém .....	7
2.2.1	Databáze .....	7
2.2.2	Systém řízení báze dat .....	7
2.3	Relační databázový model .....	8
2.3.1	Klíče .....	9
2.3.2	Vztahy .....	10
2.3.3	Typ vztahů .....	10
2.3.4	Integrita dat .....	12
2.4	Metodologie návrhu databáze .....	12
2.4.1	Konceptuální návrh databáze .....	13
2.4.2	Logický návrh databáze .....	14
2.4.3	Fyzický návrh databáze .....	18
2.5	Jazyk SQL .....	19
2.6	Proces datové analýzy .....	20
3	Analýza současného stavu .....	22
3.2	Nástroj analýzy .....	24
4	Návrh a implementace databázového systému .....	27
4.2	Konceptuální návrh .....	27
4.3	Logický návrh databáze .....	30
4.3.1	ER diagram .....	30
4.3.2	Tabulky .....	31
4.4	Fyzický návrh databáze .....	38
4.4.1	Vztahy mezi tabulkami .....	39
4.4.2	Uložené procedury .....	43

5	Závěr.....	44
	Seznam použité literatury .....	45
	Seznam zkratek	
	Prohlášení o využití výsledků mé bakalářské práce	
	Seznam příloh	

# 1 Úvod

Informační technologie slouží jako podpora ve velkých firmách již řadu let. V poslední době se však stále více rozmáhá a její vliv roste také ve středních a malých podnicích. Stupeň zavedení IT do podniku se v dílčích řešeních liší. Může jít o zastřešení pouze dílčích částí podnikových činností, nebo veškerých podnikových agend jako celku. Nový informační systém může být vytvořen z modulů zaměřujících se vždy na určitou problematiku, jako například účetnictví, mzdy, skladové hospodářství, nebo může být zakoupen jako celek, tedy hotové řešení. Na trhu je hotových softwarových produktů ať už pro drobné podnikatele nebo velké společnosti nepřehledné množství. Je však otázkou, nakolik tato hotová řešení skutečně vyhovují a podporují firemní procesy v konkrétním podniku. Proto se nabízí také třetí alternativa pro zavedení nového IS v podniku a to jeho vytvoření na míru, buď v rámci IT oddělení firmy anebo například formou outsourcingu externí firmou.

Bakalářská práce se zabývá analýzou vybrané společnosti, navržením optimálního databázového řešení na základě poznatků a požadavků plynoucích z analytické fáze a implementací návrhu. Práce je rozdělena do části teoretické a části praktické. V teoretické části se kapitola nazvaná *Teoretická východiska návrhu databázového systému* zabývá obecným popisem principu databází, jako jsou systém řízení báze dat nebo relační databázový systém. Kapitola se dále zabývá základními pojmy, jako jsou entity, atributy, klíče, vztahy a referenční integrita. V podkapitolách nazvaných *Konceptuální návrh databáze*, *Logický návrh databáze* a *Fyzický návrh databáze* jsou popsány kroky jednotlivých fází návrhu, dle kterých je pak postupováno při samotném návrhu databázového systému. Je zde také nastíněna problematika normalizace. V závěru kapitoly je pak popsán programovací jazyk SQL, server Microsoft SQL a neméně důležitý proces provádění analýzy.

V kapitole *Analýza současného stavu* se již přistupuje k praktické části a kapitola se věnuje popisu firmy a procesů ve firmě, pro kterou je systém navrhován. Analytická část bude provedena pomocí nástroje procesní analýzy *Data Flow Diagram*, o kterém je v teoretické rovině v této kapitole také zmíněno. Další popis současné situace je slovní a vychází z rozhovoru se zaměstnanci firmy a zároveň budoucími uživateli navrhovaného systému.

Závěrečná kapitola je již věnována samotnému návrhu databáze, který vychází z analýzy popsané v třetí kapitole. *Návrh a implementace databázového systému* je strukturován do podkapitol shodně s fázemi návrhu databázového systému. V konceptuální fázi je prioritou



entitně-relační diagram zahrnující slovník dat, z kterého pak vychází logická fáze návrhu. V této podkapitole jsou navrženy a popsány jednotlivé tabulky a definovány důležité primární a cizí klíče.

Poslední fází a podkapitolou je fyzický návrh databáze. Zde probíhá již samotná implementace navržených tabulek v SQL serveru, vytvoření databázového diagramu a s tím souvisejícím propojením tabulek a dále vytvoření uložených procedur pro vkládání nových údajů, mazání a úpravu dat v tabulkách.

V závěru celé práce je shrnutí a zhodnocení průběhu návrhu databázového systému, a zhodnocení dosažení stanovených cílů.

## **2 Teoretická východiska návrhu databázových systémů**

### **2.2 Databázový systém**

#### **2.2.1 Databáze**

Pro sběr, správu a kontrolu dat, které jsou používány a vytvářeny v organizacích, nám slouží informační systém, který dále umožňuje transformaci dat na informace. Jednou z nejdůležitějších složek jádra takového systému je databáze, která ho podporuje. Dnešní princip databáze však vznikl postupně. Dřívější systém použití databáze a práce s daty nebyl efektivní, protože při velkém množství dat a dat o datech byl problém se v těchto datech orientovat a hledat potřebné informace.

Pro získání informace z těchto dat je třeba tato data zpracovávat. Při hromadném zpracování dat však vznikalo mnoho problémů, jako je např. redundance dat, kdy se v různých souborech objevovala stejná data. Dále nekonzistentnost, což úzce souvisí s redundancí, kdy např. dvě kopie dat nejsou stejné a mají jinou vypovídací hodnotu. Data jsou navzájem závislé, jsou problémy s přístupem více uživatelů a ochranou dat. Problémy souborového přístupu k datům vedlo ke vzniku a rozvoji databázového systému.

Úkolem většiny databází je modelovat určitý aspekt reálného světa. Tato modelovaná část může být ve své podstatě chaotická a komplikovaná, je proto důležité omezit návrh systému na konkrétní, dobře definovanou množinu objektů a jejich vzájemných vztahů; jen tak můžeme provádět rozumná rozhodnutí ohledně záběru výsledného systému. (Šimonová Stanislava, 2007) Databázový přístup se od souborového liší tak, že nespojuje uživatelské programy s definicí a údržbou dat. Některé výhody databázového přístupu zahrnují kontrolu nadbytečnosti dat, konzistence dat, sdílení dat, vylepšené zabezpečení a integritu. Některé nevýhody zahrnují složitost, náklady, snížený výkon a vyšší dopad selhání. (Conolly, Holowczak, & Beqq, 2009)

Data v databázi jsou integrována s minimálním množstvím duplikací a jsou sdílena serverem společnosti, která ji využívá. Tato data jsou popsána. Popis těchto dat, nebo-li metadata, je také nazýván jako slovník dat. Ten je součástí databáze a přístup k němu má SŘBD.

#### **2.2.2 Systém řízení báze dat**

Systém řízení báze dat (dále SŘBD) je software sloužící k přístupu k datům, a umožňuje uživatelům tato data měnit, aktualizovat, vkládat, vyvolávat, a poté z nich vyhodnocovat.

SŘBD poskytuje kontrolovaný přístup k databázi. Poskytuje zabezpečení, integritu, řízení souběžného přístupu a zotavení a uživatelsky přístupný katalog. Poskytuje také mechanismus pohledů, aby zjednodušil data, kterými se uživatelé musí zabývat. Všechny přístupy k databázi probíhají pomocí SŘBD. (Conolly, Holowczak, & Beqq, 2009)

Databázový systém je tedy složen z databáze a ze systému řízení báze dat. Pro funkci databáze je potřeba databázové aplikace, která ji používá.

### 2.3 Relační databázový model

Model může být reprezentace, nebo abstrakce reálných objektů, událostí a jejich souvislostí. Modelem dat je souhrn pojetí pro popis dat v organizaci. Dále pro popis omezení a vztahů mezi těmito daty. Má tak za úkol reprezentovat datové požadavky organizace. Systémy implementující relační databázový model jsou pak relační databázové systémy, nebo relační databáze. Relační model, jenž je založen hlavně na matematických principech teorie množiny a predikátové logice, umožňuje snadné čtení dat, zajišťuje přesnost a konzistenci dat a poskytuje databázovou strukturu nezávislou na aplikacích provádějících přístup k uloženým datům. (Sheldon, 2005) Charakteristické pro tyto systémy je realizace vazeb pomocí tabulek – relací.

Relační model se skládá z pěti hlavní složek.

- Relace – cokoliv, co je uspořádáno do tabulky, která má sloupce a řádky a obsahuje skalární hodnoty. Řádky v tabulce odpovídají datovým n-ticím a sloupce pak atributům.
- Datová n-tice – datovové n-tice u atributů se mohou zobrazit v různém pořadí, stále půjde o stejnou relaci.
- Atribut – název sloupce relace.
- Doména – každý atribut je spojen s doménou, která může být pro každý atribut jedinečná, nebo také může být spojena s více atributy. Doména umožňuje definovat význam a zdrojové hodnoty atributu.
- Relační databáze – tabulky, které jsou na základě normalizace strukturovány.

(Conolly, Holowczak, & Beqq, 2009, str. 63)

V knize Mistrovství – databáze od spisovatelů Conolly, Begg a Holowczak vydané v roce 2009 je jednou z definic, že *relace je množina smysluplných spojení mezi entitami*. Relace je však sada sloupců a řádků seskupených do struktury podobné tabulce, která reprezentuje

jedinou entitu tvořenou ze souvisejících dat. (Sheldon, 2005) S definicí a faktem, že relace je vztah, se člověk může setkat nejen ve spoustě publikací, ale také například v počítačovém programu Microsoft Office Access. Může tomu být pro to, že z anglického *relation* se překládá do češtiny jako *vztah*.

Relační tabulky mají jisté vlastnosti. Název relace nebo tabulky má jedinečné jméno v rámci příslušné databáze. Každý sloupec a také každý záznam musí být odlišné od všech ostatních sloupců a záznamů. Tabulky neobsahují opakující se skupiny dat, nebo-li každá buňka obsahuje přesně jednu hodnotu. Pořadí sloupců a záznamů nemá teoreticky význam, ale v praxi se může správné pořadí ukázat jako efektivnější. (Conolly, Holowczak, & Beqq, 2009)

Relační systém řízení báze dat (RDBMS) je software druhé generace DBMS. Je založen na relačním modelu dat a umožňuje vytváření, udržování, modifikování a správu relační databáze.

### 2.3.1 Klíče

Spolehlivost vztahů mezi dvojicemi tabulek, korektnost struktury tabulek a minimalizace redundantních dat je zajištěna správnou definicí klíče pro tabulku, což je jeden z nejdůležitějších prvků. Prostředkem pro přesnou identifikaci jednotlivé instance je právě klíč. Zajišťuje jedinečnost dat, což je, jak je zmíněno výše, jedna z vlastností relační tabulky a také to, že hodnoty polí pro vytvoření vztahu mezi tabulkami si navzájem odpovídají. Následující odstavce popisují jednotlivé typy klíčů.

Každá tabulka musí mít aspoň jednoho kandidáta na klíč, tedy jedno, nebo více polí, které jednoznačně identifikují jednotlivé instance entity tabulky. Klíč, který bude z kandidátních vybrán, je nazýván *primárním klíčem*, zbylé klíče jsou pak nazývány *alternativní klíče*. Pole, které má být stanoveno jako kandidátní klíč, musí splňovat určité vlastnosti.

- Hodnota primárního klíče jednoznačně identifikuje hodnoty.
- Jako identifikátor není vhodné použít vícesložkové pole.
- Nepřítomnost hodnoty, tedy hodnota null, je pro kandidátní klíč nepřijatelná.
- Kandidátní klíčem by neměly být informace pro firmu důvěrné a také informace se často měnící. (Hernandez, 2003)

Tabulka má vždy pouze jeden primární klíč a v případě podmnožinové tabulky jej může sdílet.

Objeví-li se sloupec, mající hodnotu primárního klíče ve více tabulkách, je to obvykle představa vztahu (vztahy mezi tabulkami jsou rozebrány v následující podkapitole) mezi záznamy tabulek. Sloupec, který není primárním klíčem v dané tabulce, právě odkazuje na primární klíč ve své tabulce. Takový sloupec je nazýván cizí klíč a odpovídá kandidátnímu klíči některé případně stejné tabulky.

### 2.3.2 Vztahy

Mezi dvěma či více tabulkami, entitami, existuje vztah, který vyjadřuje vazbu. Vyjadřuje informaci, kterou nelze vyvodit z atributů jednotlivých entit a je zásadní složkou relační databáze. Díky vztahům můžeme najednou zobrazit data z několika tabulek – vytváří spojení mezi dvěma logicky souvisejícími tabulkami. Relace je množina smysluplných spojení, nebo-li množina vztahů, mezi entitami. Podle toho, kolik entit je v relaci zúčastněno, se určuje *stupeň relace/vztahu*. Relace může být stupně jedna, ta je pak nazývána jako *rekurzivní*. U takové relace se objevuje tatáž entita, pokaždé však v jiném postavení. Jsou-li účastníkem entity dvě, vztah je druhého stupně a je nazýván *binární*. Binární relace je nerozšířenější typ. Relace ternární a vyšší, jsou *relace složitě*, a jsou to ty, které mají ve vztahu více než 2 entity.

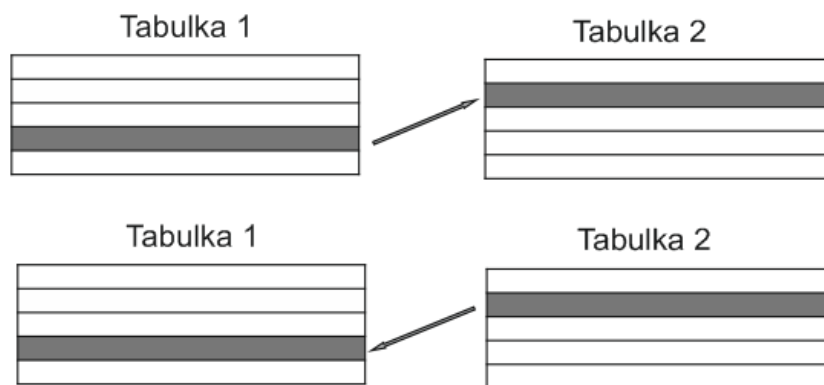
Nesprávné zřízení vztahů se může po delší době projevit jako velká nepříjemnost, proto je třeba dodržovat integritu na úrovni vztahů, správně tak definovat vztahy a také správně určit jejich typ.

### 2.3.3 Typ vztahů

Jak už bylo zmíněno, vztah je spojení dvou nebo více tabulek. Toto spojení však může být trojího typu – *kardinality*. Jaký typ spojení je mezi tabulkami, závisí na tom, jaký význam a vypovídací hodnoty má pro nás vztah mezi těmito entitami. Což nepřímo souvisí s tím, zda má tabulka primární klíč a je nezávislá na jiné tabulce, nebo je-li slabou entitou (je závislá svým primárním klíčem na existenci jiné entity).

#### Kardinalita 1:1

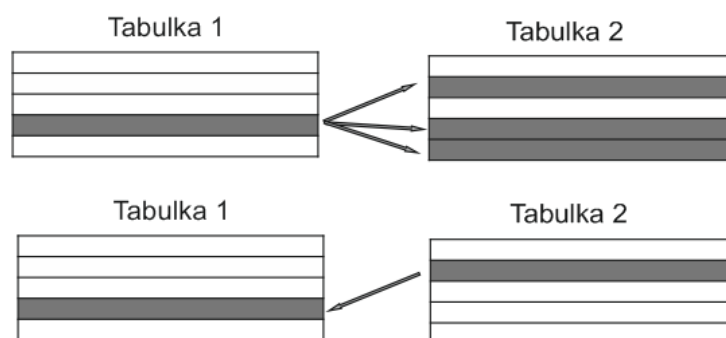
Dvojice tabulek je k sobě svázána vztahem typu 1:1, pokud je každý záznam v první tabulce svázán s právě jedním záznamem druhé tabulky a zároveň každý záznam z druhé tabulky je svázán s právě jedním záznamem z první tabulky. (Hernandez, 2003, str. 227)



Obr 1: Tabulky ve vztahu 1:1 z pohledu tabulky Tabulka 1 a z pohledu tabulky Tabulka 2 (Hernandez, 2003)

### Kardinalita 1:N

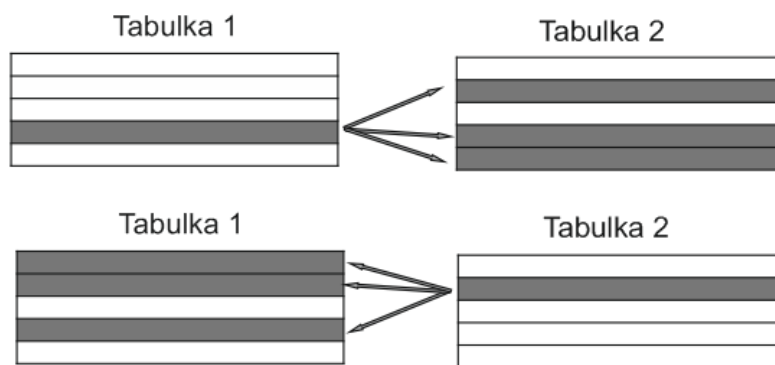
Protože jeden atribut entity z tabulky první (Odběratel) může být spojen s více atributy entity z tabulky druhé (Objednávka přijatá), avšak záznam z druhé tabulky (Objednávka přijatá) je navázán pouze s jedním záznamem z tabulky první (Odběratel), nazýváme tento vztah 1:N. Toto je nejčastější typ vztahu a nejjednodušěji se rozpozná.



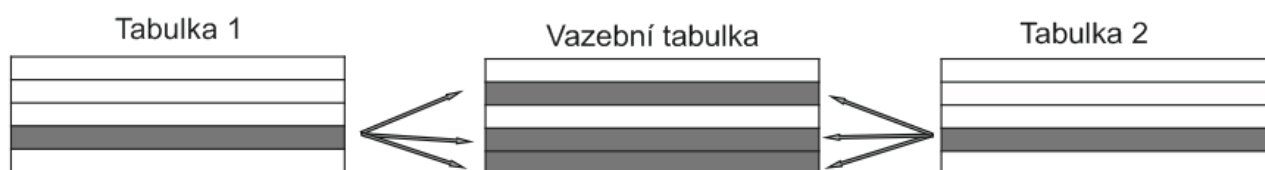
Obr 2: Tabulky ve vztahu 1:N z pohledu tabulky Tabulka 1a z pohledu tabulky Tabulka 2 (Hernandez, 2003)

### Kardinalita M:N

Tento vztah nastává, když atribut entity z tabulky první je svázán s více atributy z tabulky druhé a stejně tak opačně. Tento vztah je složitější a způsobuje, že jedna z tabulek může obsahovat velké množství redundantních dat a jedna z tabulek bude mít data duplicitní. V takovém případě zřizujeme tzv. vazební tabulku, která daný problém řeší tím způsobem, že mezi původními tabulkami už není přímý vztah, a mezi novou tabulkou a stávajícími tabulkami je vztah 1:N.



Obr 3: Vztah M:N z pohledu tabulky Tabulka 1 a z pohledu tabulky Tabulka 2 (Hernandez, 2003)



Obr 4: Použití vazební tabulky ve vztahu M:N (Hernandez, 2003)

### 2.3.4 Integrita dat

Integrita dat je jeden z nejdůležitějších aspektů při návrhu databáze a rozumí se tím platnost, přesnost a konzistentnost dat v databázi. Všechny typy integrity se implementují v průběhu návrhu databáze. S jednou z nich jsme se setkali v podkapitole věnované relacím, tedy integrita na úrovni vztahů – referenční integrita. Tato podmínka má zajistit synchronizaci dat ve vzájemně propojených tabulkách, a to splněním podmínek o správném určení účasti každé tabulky, stupně vztahu a správném zřízení vztahu. Integrita entity zajišťuje neexistenci duplicitních dat a fakt, že primární klíč musí být jedinečný a neobsahovat hodnotu null. Dále, že tabulka neobsahuje vícehodnotová, vícesložková a vypočítaná pole. Další z integrit je na úrovni pole a zabezpečuje spolehlivost polí, ve smyslu přesnosti dat v polích, a že pole stejného typu jsou v celé databázi definována konzistentně. V případě, že v databázi jsou použity pohledy, je také důležité prověřit jejich správnost.

## 2.4 Metodologie návrhu databáze

Při návrhu databáze vycházíme z analýzy, která je popsána v následující kapitole. Navrhujeme-li složitější databázi, prochází tento proces určitými stádii a kroky. Tyto stádia mohou být různá, avšak v případě této práce a tohoto návrhu databáze jsou následující: konceptuální návrh databáze, logický návrh databáze a fyzický návrh databáze. Jednotlivé

fáze používají procedury, techniky, nástroje a dokumentaci, a pomáhají tak návrháři při plánování a kontrole projektu. Podporují a usnadňují tak celý proces návrhu.

### **2.4.1 Konceptuální návrh databáze**

Vychází z volného slovního popisu objektu. Konceptuální model dat je vytvářen z dat bez jakýchkoli podrobností, jak by tomu mohlo být v případě, vycházel-li by z například relačního datového modelu. V některých zdrojích informací však může být uvedeno, že vychází ze sémantického modelu dat, což je fáze zahrnující strukturovaný popis datové části objektivní reality řešeného projektu IS. Identifikuje tak důležité entity a relace na základě informací získaných z analýzy.

Ke grafickému znázornění modelu dat se používá *ER diagram*, který je součástí ER modelu, *Entity-relationship model*. Metoda datového modelování, konceptuální model vysoké úrovně, který představuje úplnou a přesnou reprezentaci datových požadavků organizace, kterou má databáze podporovat. (Conolly, Holowczak, & Beqq, 2009) ER model zahrnuje kromě diagramu, jež pomocí symbolů a diagramů tvoří srozumitelné schéma databáze, také slovník dat a obsahuje: entity, relace, atributy a domény atributů, primární a alternativní klíče, integritní omezení. K vytvoření ERM vede několik kroků, prvním je identifikace entity. Entity pro model získáme specifikací uživatelských požadavků především slovníku dat. Jak bylo zmíněno v předchozích kapitolách, mezi entitami existují vztahy. Identifikace vztahů je v ER modelu neméně důležitá a vychází také především z požadavků uživatelů. Zde patří také určení a omezení multiplicity, což slouží ke kontrole a udržení kvality dat.

Následujícím třetím krokem tvorby modelu v konceptuální úrovni je otázka, jaké informace budeme o jednotlivých entitách a relacích uchovávat. Jedná se tedy o identifikaci atributů a i zde se bude čerpat především ze slovníku dat, který obsahuje uživatelské požadavky. Při identifikaci atributů je vedena dokumentace, která zachycuje pro uživatele jasně srozumitelné názvy, jejich popis, typ a délku dat. Dále může-li se vyskytovat hodnota null a je-li atribut vícehodnotový. Dříve, než se určí kandidátní a následně primární, případně alternativní klíče, jsou určeny domény atributu. Doména je množina hodnot, z níž mohou čerpat své hodnoty jeden nebo více atributů. (Conolly, Holowczak, & Beqq, 2009, str. 218) Přezkoumáním relací typu jedna k jedné a odstraněním relací nesoucích stejné informace se provede krok kontroly redundance modelu.



V kapitole páté, tj. samotném návrhu databáze, je tento krok nutné provést, protože v ER diagramu se objeví hned několik relací jedna k jedné. V sedmém kroku návrhu kompletního ER modelu se provede kontrola podpory uživatelských transakcí. Tato kontrola je prováděna manuálně nad ER diagramem a slovníkem dat, a to pomocí popisu transakce nebo sledování cest transakcí. Posledním krokem před kompletním dokončením konceptuálního návrhu je posouzení ER diagramu a veškeré dokumentace, která dohromady dává ER model, s uživateli. V případě provádění změn je možnost návratu k předcházejícím krokům. Konceptuální návrh je zdrojem informací pro logický návrh databáze, který je popsán v následující kapitole. (Conolly, Holowczak, & Beqq, 2009)

### **2.4.2 Logický návrh databáze**

Druhý krok metodologie návrhu databáze mapuje ER model, popsáný v předchozí kapitole, do množiny tabulek a kontroluje, zda tabulky splňují požadavky uživatelů. Tento model již vychází z relačního modelu dat. Jedním velkým krokem celého návrhu databáze a stěžejním krokem logického návrhu je mapování ER modelu do tabulek. Ten se pak, stejně jako tomu je v konceptuálním návrhu, skládá z několika menších kroků. Prvním z nich je vytvoření množiny tabulek podle ER modelu. Tabulky reprezentují entity, vztahy, atributy a integritní omezení. Struktura tabulky je dána ER diagramem, slovníkem dat a další již vytvořené dokumentace. V kapitole 2.5 Jazyk SQL jsou popsány další jazyky, které SQL zahrnuje. Jedním z nich je DDL – Data definitiv language a je použit právě pro popis obsahu tabulek.

Druhým krokem návrhu je kontrola struktury tabulek pomocí normalizace.

#### **Normalizace**

Myšlenka normalizace je proces organizace databáze do struktury, která zachovává integritu a konzistentnost dat. Normalizace je základním principem relačního modelu a provádí se často jako řada testů na tabulce. Zkoumá se dodržení nebo porušení pravidel pro normální formu. Normální formy jsou založeny na vztazích mezi sloupci tabulky a určující pravidla, jak by data měla být organizována, aby se zamezilo nekonzistenci a ztrátě dat při používání databáze. Normálních forem existuje několik, původní relační model obsahoval tři a ty jsou stále nejužívanějších z nich a zároveň formy použité v této práci. Jsou to první normální forma (1NF), druhá normální forma (2NF) a třetí normální forma (3NF).

Tabulky s redundantními daty mohou trpět problémy nazývanými anomálie aktualizace, které jsou dále klasifikovány jako anomálie vkládání, vymazání nebo modifikace. Tyto anomálie se projevují ztrátou nebo nekonzistencí dat.

#### **a. První normální forma**

Tabulka, v níž každý průsečík sloupce a záznamu obsahuje jen jedinou hodnotu. (Conolly, Holowczak, & Beqq, 2009) V knize SQL začínáme programovat napsané Robertem Sheldonem a vydané v roce 2005 je první normální forma definována takto:

- každý atribut skupiny musí obsahovat pouze jednu hodnotu,
- každá skupina v relaci musí obsahovat stejný počet hodnot,
- každá skupina v relaci musí být odlišná.

1NF je základem pro druhou a třetí formu a je jedinou formou, která je kriticky důležitá pro vytvoření vhodných tabulek pro relační databáze.

#### **b. Druhá normální forma**

Tabulka, která je v 1NF a ve které jsou hodnoty každého sloupce, který není součástí primárního klíče, determinována všemi hodnotami sloupců, které tvoří primární klíč. (Conolly, Holowczak, & Beqq, 2009, str. 192)

Druhá normální norma určuje, že relace musí být v první normální formě a že všechny atributy v relaci jsou závislé na celém klíči kandidáta. (Sheldon, 2005, str. 26) Co je kandidátní klíč, je popsáno v předchozí kapitole. Tato forma se týká jen tabulek, jejichž primární klíč tvoří dva nebo více sloupců, tedy mající složený primární klíč – nemá-li tabulka složený primární klíč, a splňuje-li první normální formu, splňuje tak zároveň i formu druhou.

V druhé definici je zmíněna *závislost* na klíči kandidáta. Funkční závislost je mechanismus, který určuje závislost hodnoty atributu na hodnotě atributu jiného (případně na hodnotách skupiny atributů). V opačném směru funkční závislost platit nemusí. (<http://www.biexperts.cz/index.php/cs/professional/18-ctsql/31-arnormalization.html>) Jsou-li v tabulce dva primární klíče a na každém z nich jsou závislé jiné neklíčové atributy, tabulka není v 2NF. Tato situace se nazývá částečnou závislostí a řeší se dekompozicí relačního schématu, což znamená, že např. jednu tabulku rozložíme na dvě. Částečná závislost také často způsobuje redundantní data v tabulce, vzniká anomálie aktualizace a při změně jednoho záznamu, musíme provést dvě aktualizace tabulky. Data by jinak byla nekonzistentní.

### c. Třetí normální forma

Třetí normální forma, stejně jako druhá normální forma, je závislá na klíči relace. Pro splnění třetí normální formy musí být každá relace v druhé normální formě a atributy, které nejsou klíči, nesmí být závislé na ostatních a musí být závislé na klíči. (Sheldon, 2005, str. 26)

Tabulka, která již je v 1NF a 2NF a ve které všechny hodnoty ve sloupcích, které nepatří k primárnímu klíči, jsou determinovány pouze sloupci primárního klíče a nejsou determinovány žádnými jinými sloupci. (Conolly, Holowczak, & Beqq, 2009, str. 195)

Podobně, jako v 2NF způsobuje částečná závislost redundanci dat, také v 3NF existuje anomálie aktualizace způsobená závislostí – tranzitivní závislostí. Tranzitivní závislost popisuje vztah, kdy jsou neklíčové atributy determinovány jinými atributy. V knize Mistrovství – databáze vydané v roce 2009 je uvedena tato definice: „Tranzitivní závislost popisuje vztah mezi sloupci  $a$ ,  $b$ , a  $c$ . Pokud  $a$  determinuje  $b$  a  $b$  determinuje  $c$ , pak sloupec  $c$  je tranzitivně závislý na  $a$  prostřednictvím  $b$  (pokud  $b$  nebo  $c$  nedeterminuje  $a$ ).

Obecně je-li  $a$  závislé na  $b$ , pak  $b$  je determinováno  $a$ .

### d. Boyce/Coddova normální forma

Boyce-Coddova normální forma je původní definicí 3NF a nyní je jakousi její „variací“. Je významná proto, že udává i pravidla uvnitř složeného primárního klíče.

Definici Boyce-Coddovy normální formy říká, že relace  $R$  je v Boyce-Coddově normální formě, pokud pro každou funkční závislost  $X \rightarrow Y$ , kde  $X$  a  $Y$  jsou množiny atributů a zároveň kde  $Y$  není podmnožinou  $X$ , platí, že  $X$  je nadmnožina nějakého klíče, nebo  $X$  je klíčem relace  $R$ . Boyce-Coddova normální forma se tedy týká především takových relací, které mají více klíčů. Protože definice může být poněkud nesrozumitelná, pokusíme se tuto normální formu blíže vysvětlit. Aby ji relace splňovala, musí platit oba následující body:

- nesmí se stát, že by levá strana kterékoliv závislosti neobsahovala klíč relace (nebo jeho součást),
- žádné dva klíčové atributy (žádné dvě součásti klíče) by na sobě neměly funkčně záviset (to plyne z požadavku  $Y$  není podmnožina  $X$ ). (<http://www.dbsvet.cz/view.php?cisloclanku=2008102302>)

## *Denormalizace*

Za účelem například snížení složitosti implementace nebo zvýšení rychlosti zpracování dotazu je možné porušit pravidla normalizace – denormalizace. Především druhé a třetí formy. Toto porušování se však musí provádět velmi opatrně, protože proces normalizace znamená zajištění integrity dat, bez které efektivita databáze značně klesá. (<http://www.biexperts.cz/index.php/cs/professional/18-ctsql/31-arnormalization.html>, Sheldon, 2005) Musí se tedy zkontrolovat, zda všechny tabulky vytvořené v předchozím kroku jsou nejméně ve třetí normální formě. Nesplňovaly-li by tabulky 3NF, model dat nebo tabulky by se musel restrukturovat. Zde je na zvážení také jistá míra denormalizace, popsané také v úvodu kapitoly 2.2. Pro ověření, že tvorba tabulek proběhla správně, se ještě zkontroluje podpora uživatelské transakce. Prozkoumáním datových požadavků transakcí se zjistí, že všechna data jsou přítomna v jedné nebo více tabulkách. Jsou-li data ve více tabulkách, proběhne kontrola, že se dají spojit mechanismem primární/cizí klíč.

Předposledním krokem logického návrhu a tvorby tabulek je kontrola integritního omezení. Integritní omezení zabraňují tomu, aby se databáze stala nekompletní, nepřesnou nebo nekonzistentní. V této fázi návrhu je důležitý jen návrh vysoké úrovně, který popisuje, jaká integritní omezení jsou požadována, bez ohledu na to, jak je možné jich dosáhnout. Integritní omezení může být vícero. Je to např. omezení na požadovaná data – tedy, že sloupce smí nebo nesmí obsahovat hodnotu null. Identifikace domén atributů probíhá již v konceptuálním návrhu, ve fázi logického návrhu je však ještě kontrolována. Například v případě návrhu, který je popisován v této práci by domény atributů mohly být v tabulce Odběratel – Název odběratele a omezení na názvy odběratelů.

Dalším integritním omezením je integrita entit, kde je stanoveno, že primární klíč nesmí obsahovat hodnoty null. Mezi omezeními nesmí chybět také velmi důležitá referenční integrita – obsahuje-li cizí klíč hodnotu, pak tato hodnota musí odkazovat na existující záznam v rodičovské tabulce. Existují určitá existenční omezení, která zajišťují referenční integritu, a to definováním podmínek, kdy je možné vložit, aktualizovat nebo vymazat primární nebo cizí klíč. V závěru logického návrhu a tvorby tabulek je zapotřebí posoudit logický návrh s uživateli databáze.

### 2.4.3 Fyzický návrh databáze

Výchozím bodem této kapitoly je logický model dat a dokumentace, která model dat popisuje. V úvodu fyzického návrhu je zapotřebí srovnat logický a fyzický návrh databáze. Zatímco v logickém návrhu je v zájmu otázka co, při fyzickém návrhu databáze je v zájmu otázka jak. „Fyzický návrh databáze je proces vytvoření popisu implementace databáze vnějších paměťových zařízení; popisuje podkladové tabulky, organizaci souborů, indexy použité kvůli efektivnímu přístupu k datům a všechna související integritní omezení a bezpečnostní omezení. Návrh podkladových tabulek je možné provést, teprve když dobře známe možnosti nabízené cílovým DBMS.“ (Conolly, Holowczak, & Beqq, 2009, str. 283)

Stěžejním krokem v konceptuálním návrhu je vytvoření ER modelu, v konceptuálním návrhu pak jeho mapování do tabulek. Ve fyzickém návrhu databáze je takových hlavních kroků více. Prvním z nich je převod logického návrhu databáze do cílového DBMS. Překlad tabulek se provádí na základě shromážděných informací získaných v předchozí fázi návrhu. Tyto informace získáme ze slovníku dat a definic tabulek v jazyce návrhu databáze. Tento jazyk se později musí rozšířit, aby bylo možné definovat také domény, implicitní hodnoty a hodnoty null. Vyskytují-li se ve slovníku dat odvozené sloupce, tedy sloupce s hodnotami zjištěných pomocí jiných sloupců, proběhne jejich shromáždění a poté posouzení, zda se takový sloupec do databáze zavede, nebo proběhne jeho odvození či vypočítání vždy až v případě potřeby.

Posledním krokem převodu z logického návrhu do fyzického je návrh všech zbývajících integritních omezení platných pro data. V logické fázi již návrh na požadovaná omezení dat, omezení domén a relační integritní omezení proběhl, ve fyzické fázi se z něj tedy vychází. Stejně jako se musí zdokumentovat zvolené řešení pro odvozené sloupce, probíhá dokumentace návrhu integritních omezení. Druhým hlavním krokem fyzického návrhu je volba organizace souborů a indexů. „Index je datová struktura, která DBMS umožňuje rychleji lokalizovat konkrétní záznamy v souboru, a tím zlepšit odezvu na uživatelské dotazy.“ (Conolly, Holowczak, & Beqq, 2009, str. 269) Tento krok zahrnuje analýzu transakcí (které přistupují k tabulkám), kde je cílem zjistit funkčnost transakcí a tím identifikovat ty transakce, které mohou ovlivnit výkonnost systému. Analyzují se často jen ty transakce, které jsou často používány, nebo ty které jsou pro provoz organizace kriticky důležité. Transakce přistupují k tabulkám, proto je v dalším kroku nutné zvolit efektivní organizaci souborů pro všechny podkladové tabulky, protože jedním z hlavních cílů fyzického návrhu databáze je

efektivní uložení dat. Je-li zapotřebí zvýšení výkonnosti systému a efektivnější vyvolání dat je možné zvážit volbu sekundárních indexů.

V první fázi metodologie návrhu databáze jsou navrhovány uživatelské pohledy dle analýzy a požadavků uživatelů. V třetím hlavním kroku poslední fáze návrhu je cílem návrh právě takovýchto uživatelských pohledů. Tento krok je důležitý zvláště v případě víceuživatelských SŘBD. Dle analýzy a požadavků uživatelů jsou v dalším kroku návrhu navrženy bezpečnostní mechanismy. Relační databázové systémy obecně poskytují dva typy bezpečnostních opatření: zabezpečení systému, zabezpečení dat. „Zabezpečení systému se týká přístupu a používání databáze na systémové úrovni, jako například uživatelských jmen a hesel. Zabezpečení dat se týká přístupu k objektům databáze (jako tabulkám a pohledům) a činností, které mohou uživatelé s těmito objekty vykonávat.“(Conolly, Holowczak, & Beqq, 2009, str. 283).

V kapitole Normalizace dat 2.2.1 je jako jisté řešení zmíněna denormalizace. V tomto kroku návrhu je důležité její zvážení. Je-li výkon nedostatečný a tabulka není často aktualizovaná a zároveň často dotazovaná, je volba denormalizace přijatelná. Při denormalizaci dat je třeba zvážit její důsledky a to, jak bude udržována integrita dat. Je-li denormalizace zavedena, je na místě dokumentovat redundanci a aktualizovat logický model dat. V této fázi začíná být systém plně v provozu, proto je na řadě jeho monitorování a případné zlepšení výkonnosti, opravy nevhodného návrhu nebo zohlednění změněných požadavků. Správu a kontrolu fyzické realizace databázového systému, včetně fyzického návrhu databáze a jeho implementace má na starost administrátor databáze a je to obecně záležitost správy databáze.

## **2.5 Jazyk SQL**

Structured Query Language – strukturovaný dotazovací jazyk byl vyvinut v letech 1974 – 1979 a dnes je z něj nejrozšířenější databázový jazyk a díky standardům ANSI a ISO je nyní de facto jazykem pro definici a manipulaci relačních databází. (Conolly, Holowczak, & Beqq, 2009) Jde o neprocedurální jazyk, což znamená, že nevyžaduje určení přístupové metody k datům. Jednou z hlavních vlastností SQL je, že struktura příkazů je založena na standardních anglických slovech jako SELECT (vyber), INSERT (vlož), UPDATE (aktualizuj), a DELETE (vymaž). Dotazovací jazyk SQL je nezávislý na fyzické organizování dat a je založen na relačním modelu, přestože není jeho přesnou implementací. Jazyk je zamýšlen jako ideální databázový jazyk. Ten by měl uživateli umožnit vytvářet struktury databází a tabulek. Dále provádět údržbu dat, například vkládání, modifikaci a vymazání dat

z tabulek a provádět jednoduché i složité dotazy. Jazyk by měl být přenositelný – měl by odpovídat nějakému uznávanému standardu. Standardizace organizací ANSI proběhlo roku 1986 a její přijetí ISO v roce 1987. Poslední verzí standardu je SQL:1999.

Standard SQL ISO zahrnuje jazyky DDL a DML. *Data Definition Language*, tedy jazyk pro popis a definici dat. Příkazy se používají pro vytváření, úpravu nebo odstraňování tabulek, pohledů, schémat apod. *Data Manipulation Language* je jazyk sloužící pro manipulaci s daty, např. prohlížení, přidávání nebo úpravy dat, které jsou uloženy v databázových objektech.

### *Microsoft SQL Server*

Programovat v jazyku SQL můžeme pomocí několika SŘBD. Jedním z nich je SQL server vyvinutý společností Microsoft. Tento server je výkonný relační databázový a analytický systém. Je založen na architektuře klient/server a může být využit pro řešení datových skladů.

## **2.6 Proces datové analýzy**

Při vývoji databázového systému je jedna z podstat kvalitní analýza současného stavu a definice systému včetně jeho hlavních uživatelských pohledů. Pohledů může být více a je důležité je identifikovat správně a všechny, aby nebyl opomenut žádný z požadavků na databázi. Uživatelský pohled definuje, co se od databázového systému požaduje z hlediska uchovávaných dat a prováděných transakcí s daty (jinak řečeno, co uživatel bude s daty dělat). (Conolly, Holowczak, & Beqq, 2009, str. 111) Sběrem a analýzou požadavků o organizaci dosáhneme informací k určení požadavků na nový databázový systém. Pan Michael J. Hernandez ve své knize Návrh databází uvádí, že proces analýzy má tři základní kroky: určování způsobu získávání dat, určování způsobu prezentace dat a provádění rozhovorů s uživateli a managementem.

Prvním krokem, určení způsobu získávání dat se rozumí způsob, jakým si daná organizace shromažďuje data. Zkoumají se záznamy na papírech, typy papírových dokumentů, počítačové programy používané pro sběr dat a používá-li firma webové stránky, i zde se získaná data prozkoumají. Druhým krokem je určování způsobu prezentace informací, tedy to jak organizace převádí získaná data na informace. Můžou to být například tyto způsoby prezentace: *Zprávy*. Textové či tabulkové dokumenty, které podávají uspořádáním dat informace o těchto získaných datech. *Prezentace*. Organizovaně uspořádané snímky zahrnující zpracovávána data promítány na počítači. Posledním typem prezentace dat jsou *webové stránky*. Ve chvíli, kdy je obecná představa o tom, jak organizace data získává a

prezentuje, je čas zjistit, jak jsou tato data používána. Potom také získat pomocné informace pro definici předběžných struktur polí a tabulek a získat budoucí požadavky na informace. To vše pomocí rozhovorů s uživateli a managementem firmy.



### 3 Analýza současného stavu

Firma, pro kterou je navrhován tento databázový systém se nazývá MORATON. Sídli ve Frenštátě pod Radhoštěm a vznikla roku 2005. Firma se zabývá reexportem zboží.

Objednává různé zboží pro zákazníky ze Střední Asie. Toto zboží objednává ze zemí Evropské unie, některé zboží převážně z Jižní Evropy a některé, český porcelán a křišťál, je dováženo od našich výrobců. Nejčastějším objednávaným zbožím je tzv. bílá technika, tedy zboží jako jsou myčky nádobí, lednice, digestoře nebo také fény, holicí strojky apod. Toto zboží je objednáváno ze zemí Evropské unie na základě objednávky přijaté od odběratelů z Kazachstánu – Rollant, Mekon. Pro tyto odběratele je za měsíc posíláno průměrně pět kamiónů se zbožím, přičemž průměrně osm kamiónů přiveze zboží do firmy pro jeden kamion.

Další dva kamióny v měsíci vezou zboží pro odběratele luxusního zboží z českého porcelánu či křišťálu, například lustry, jídelní servisy, nápojové sklo. Toto zboží je dodáváno například z Karlových Varů, Kamenického Šenova, Světlé nad Sázavou, nebo Jablonce nad Nisou. Dodavatelů porcelánu má firma asi deset.

Dalším značně odebíraným druhem zboží je bazénová technika, zahradní jezírka a různá bazénová chemie s tímto související, které jsou do firmy dováženy zejména ze zemí Jižní Evropy jako je Španělsko nebo Itálie. Odběratelem tohoto zboží je Kyrgyzská firma. Toto zboží je sezonní, tedy objednáváno především na jaře, výjimečně na podzim.

Firma zaměstnává pět zaměstnanců. Dva jsou na vedoucí pozici, komunikují s odběrateli a dodavateli, domlouvají obchody a dále komunikují s řidiči kamiónu, kteří přivážejí zboží. Dva lidé jsou u firmy zaměstnáni jako skladníci, kteří pomáhají s vykládáním, nakládáním a uskladněním zboží ve skladu. Pátým zaměstnancem je asistentka, která zpracovává doklady a vyřizuje poštu.

Firma nyní veškeré faktury vytváří pomocí šablony v Excelu. S daty v počítači nyní pracují zmínění vedoucí pracovníci a asistentka. Přístup k programu je potřeba pouze z jednoho počítače.

Proces jednoho obchodu vypadá takto: Objednávku od odběratele (např. firmy Mekon) obdrží firma v elektronické podobě, nejčastěji mailem, někdy také telefonicky. Objednávka obsahuje katalogové číslo dodavatele produktu, cenu bez DPH, název produktu a počet kusů.

Asistentka přepośle objednávku dodavatelské firmě, přičemž kopii objednávky zaeviduje. Po přivezení zboží od dodavatele, se zboží vyloží ve skladu a zkontroluje s dodacím listem a fakturou. Firma pak podle původní objednávky vytvoří novou fakturu, s navýšenými cenami za uskladnění, vyřízení objednávky a cenu přepravy od dodavatele. Začíná-li mít dostatek zboží na naplnění jednoho kamiónu – asi 300 ks zboží (nebo cca 90m<sup>3</sup>), informuje o tom odběratele s dostatečným předstihem a ten vyśle kamión pro dané zboží. V jednom kamiónu, může být zboží z více objednávek. Spolu se zbožím je vezen také dodací list a faktura. Kopie faktury je také posílána v elektronické podobě.

Veškeré platby jsou prováděny přes elektronický bankovní systém. Cena za zboží je uváděna v amerických dolarech, nebo eurech. Cena za dovoz zboží z Evropy je v korunách, dovoz do Kazachstánu pak v eurech. Výpočet cen za zboží, uskladnění a dovoz je prováděn ručně.

Shrnutí požadavků na systém: Přehled odběratelů a jejich správa, správa objednávek, přehled dodavatelů a jejich správa, přehled objednaného zboží, správa přijatých faktur (pro výpočet ceny), správa přijatých dodacích listů, přehled přijatého zboží – evidence zboží na skladě, správa faktur vystavených a dodacích listů vystavených.

Shrnutí požadavků na systém: Přehled odběratelů a jejich správa, správa objednávek, přehled dodavatelů a jejich správa, přehled objednaného zboží, správa přijatých faktur (pro výpočet ceny), správa přijatých dodacích listů, přehled přijatého zboží – evidence zboží na skladě, správa faktur vystavených a dodacích listů vystavených.

Návrh databázového systému pro firmu Moraton tedy vychází z analýzy, která byla slovně popsána výše. V této kapitole je však ještě použit nástroj pro analýzu, a to data flow diagram.

## 3.2 Nástroj analýzy

Data flow diagram (diagram datových toků) slouží pro popis funkčnosti informačního systému. Přesto, že v této práci není navrhován informační systém, nýbrž jen jeho část, je použit tento model pro lepší znázornění procesů ve firmě a pro lepší odvození entit. Databázový systém bude tak zároveň možné jednodušeji rozšířit na kompletní informační systém. Hierarchie diagramu datových toků začíná kontextovým diagramem a dále se skládá z jednotlivých úrovní – začínající je nultá úroveň a další, první úroveň, je použita, je-li diagram na jednu úroveň příliš složitý.

### Kontextový diagram

Jde o model chování vnějšího systému, který zobrazuje hranice mezi systémem a vnějším světem. Na tomto diagramu je vidět, jaká data do systému vstupují a vystupují a kdo se systémem komunikuje. Bezprostředním rozkladem tohoto diagramu pak vzniká DFD nulté úrovně. (<http://zdenek2.euweb.cz/doc3/prois32a.html>)



Obr 5: Kontextový diagram



- objednávka zaslaná – generování nové objednávky,
- informace o dodavateli – operace s údaji a správa informací o dodavateli,
- dod. list. přij. – (dodací list přijatý) operace s údaji o dodaném zboží,
- FAP – (faktura přijatá) operace s informacemi o cenách zboží,
- výpočet ceny – operace s cenou za zboží přijatým a zaslaným,
- FAV – (faktura vystavená) operace s informacemi o cenách zboží,
- dod. list. vyd – (dodací list vydaný) operace s údaji o dodaném zboží,
- doprava zboží – správa údajů o řidičích, autech a jednotlivých jízdách.

#### *Datové sklady*

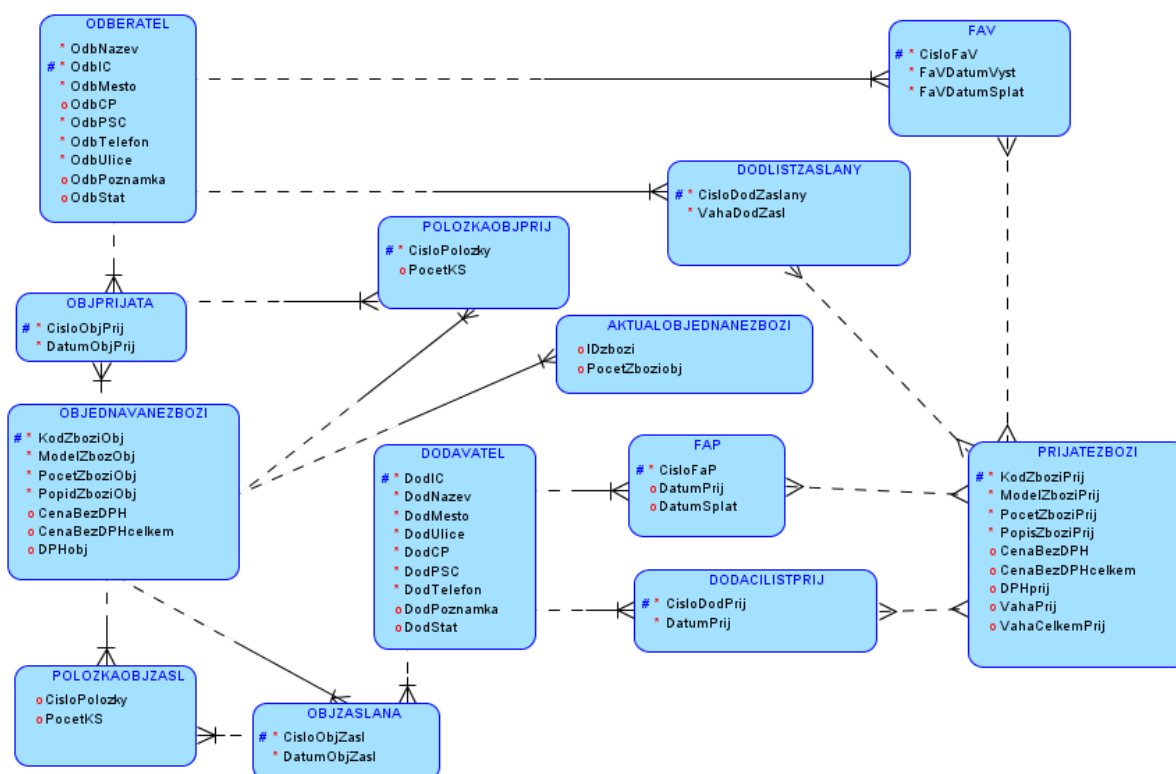
- odběratel – informace o názvu, adrese, IČ odběratele,
- objednávka – informace o objednavce a odběrateli,
- objednávané zboží – údaje o objednávaném zboží,
- objednávka zaslaná – informace o objednaném zboží, dodavateli a objednavce,
- aktuální objednané zboží – informace o momentálně objednaném zboží,
- dodavatel – údaje o názvu, adrese, IČ dodavatele,
- FAP – informace o dodavateli a cenách za objednané zboží,
- dod. list. přij. – informace o dodaném zboží,
- FAV – informace o odběrateli a cenách za objednané zboží,
- dod. list. vyd. – informace o posílaném zboží,
- dodané zboží – údaje o zboží, které je na skladě.

## 4 Návrh a implementace databázového systému

V této práci, je použita metodika návrhu pomocí třech fází: konceptuální, logický a fyzický návrh. Dle předběžných úvah, bude systém implementován v SQL Serveru.

### 4.2 Konceptuální návrh

Z analýzy podnikových procesů a funkcí vzešly entity. Jak bylo uvedeno v kapitole 4, konceptuální návrh je „oprotěn“ od uvažování o podrobnostech a pomocí ER diagramu (obrázek 7) modeluje jen entity a relace databáze.



Obr 7: ER diagram navrhovaného systému s primárními klíči a atributy

## Slovník dat

Část slovníku dat, obsahující šest entit a popis jejich atributů, datový typ atributů a možnost hodnoty null. Zbývající entity ve slovníku dat jsou v příloze č. 1.

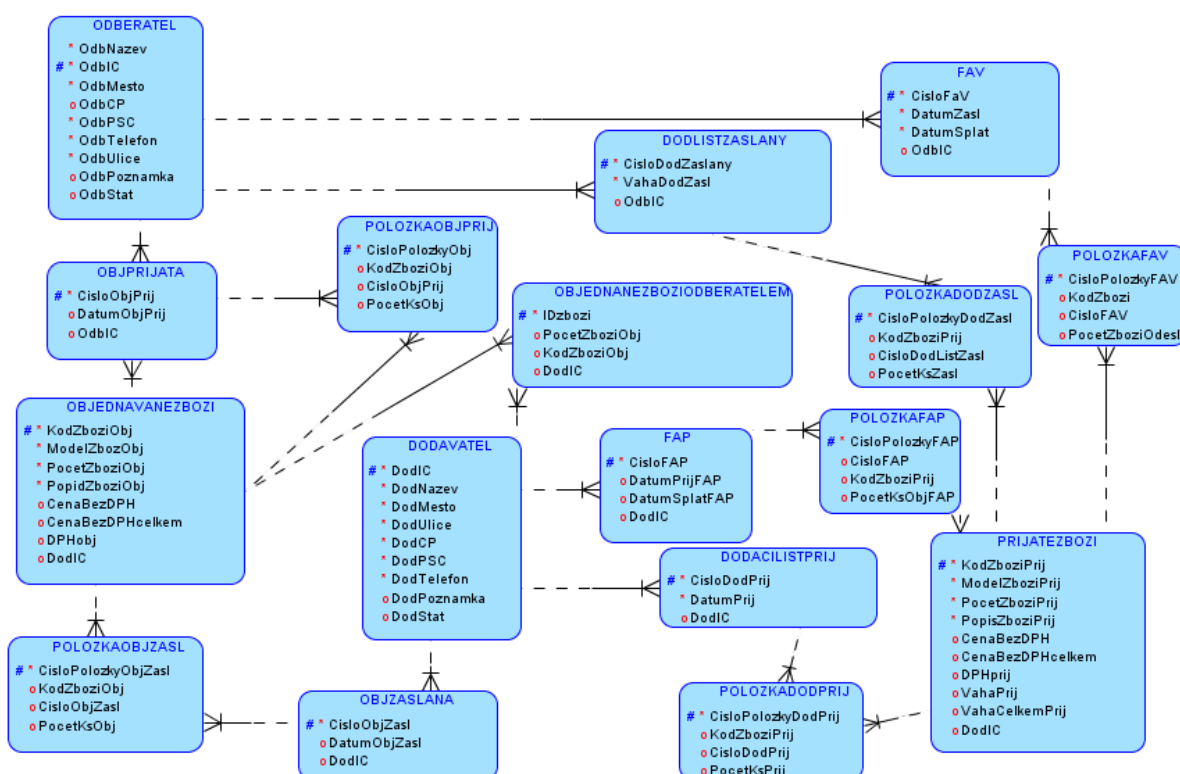
Entita	Jméno atributu	Popis	Typ a délka dat	Null
Odberatel	OdbNazev	Obchodní název odběratele	Varchar, 30	Ne
	OdbIC	IČ odběratele	Char, 8	Ne
	OdbMesto	Kontaktní adresa odběratele - město	Varchar, 30	Ne
	OdbUlice	Kontaktní adresa odběratele - ulice	Varchar, 20	Ne
	OdbCP	Číslo popisné sídla odběratele	Varchar, 10	Ne
	OdbStat	Kontaktní adresa odběratele - stát	Varchar, 20	Ne
	OdbTelefon	Telefonní kontakt na odběratele	Varchar, 15	Ne
	OdbPoznamka	Poznámka k odběrateli	Varchar, 50	Ano
Dodavatel	DodNazev	Obchodní název dodavatele	Varchar, 30	Ne
	DodIC	IČ dodavatele	Char, 8	Ne
	DodMesto	Kontaktní adresa dodavatele - město	Varchar, 30	Ne
	DodUlice	Kontaktní adresa dodavatele - ulice	Varchar, 20	Ne
	DodCP	Číslo popisné ulice odběratele	Varchar, 10	Ne
	DodStat	Kontaktní adresa dodavatele - stát	Varchar, 20	Ne
	DodTelefon	Telefonní kontakt na dodavatele	Varchar, 15	Ne
	DodPoznamka	Poznámka k dodavateli	Varchar, 50	Ano
ObjPrijata	CisloObjPrij	Číslo objednávky přijaté od odběratele	Varchar, 20	Ne
	DatumObjPrij	Datum přijetí objednávky od odb.	Date	Ne
ObjednaneZbozi	KodZboziObj	Kód objednávaného zboží	Varchar, 20	Ne
	ModelZboziObj	Model objednávaného	Varchar, 20	Ne

		zboží		
	PocetZboziObj	Počet ks objednávaného zboží	Numeric, 4,1	Ne
	PopisZboziObj	Popis objednávaného zboží	Varchar, 50	Ne
	CenaBezDPH	Cena za kus zboží bez DPH	Money	Ne
	CenaBezDPHcelkem	Celková cena za zboží bez DPH	Money	Ne
	DPHobj	Sazba daně z přidané hodnoty	Integer	Ne
ObjZaslana	CisloObjZasl	Číslo objednávky poslané dodavateli	Char, 7	Ne
	DatumObjZasl	Datum posláni objednávky dodavateli	Date	Ne



## 4.3 Logický návrh databáze

### 4.3.1 ER diagram



Obr 8: ER diagram včetně vazebních tabulek

V konceptuálním návrhu v ER diagramu je nedodržena NF. Mezi tabulkami je vztah M:N, což by nemohlo fungovat. Tento vztah byl vyřešen vazebními tabulkami: *PolozkaDodZasl*, *PolozkaFAV*, *PolozkaDodPriji* a *PolozkaFAP*.

Logická fáze návrhu vychází z relačního modelu dat. V této fázi je hlavní krok mapování ER modelu do tabulek a vztahů mezi nimi. Struktura tabulek je dána ER diagramem a slovníkem dat.

### 4.3.2 Tabulky

#### Tabulka Odberatel

Tato tabulka bude obsahovat data o odběratelích, se kterými firma pracuje nebo pracovala a nepřerušili úplně spolupráci. Primárním klíčem tabulky je *OdbIC*. Tato identifikace byla jednoznačná, protože v tabulce se další jedinečné označení nevyskytuje. Firma má za měsíc asi dva nové odběratele, přičemž s jinými dvěma přitom ukončí spolupráci. Aktuální počet odběratelů je asi pět. Primární klíč *OdbIC* se jako cizí klíč vyskytuje ve třech jiných tabulkách. Tabulka je tedy v povinném vztahu se třemi dalšími relacemi.

Název atributu	Datový typ
OdbNazev	Varchar(30)
OdbIC (PK)	Char(8)
OdbUlice	Varchar(20)
OdbCP	Varchar(10)
OdbMesto	Varchar(30)
OdbPSC	Varchar(10)
OdbStat	Varchar(20)
OdbTelefon	Varchar(15)
OdbPoznamka	Varchar(50)

Obr 9: Tabulka Odberatel

#### Tabulka ObjPrijata

V této tabulce budou jen tři sloupce. *CisloObjPrij*, což bude zároveň identifikátorem tabulky, tedy primárním klíčem. *DatumPrijObj* objednávky se bude evidovat pro kontrolu a třetím sloupcem je *OdbIC*, cizí klíč, jehož rodičovská tabulka je *Odberatele*.

Název atributu	Datový typ
CisloObjPrij (PK)	Varchar(20)
DatumObjPrij	Date
DobIC (FK)	Char(8)

Obr 10: Tabulka ObjPrij

#### Tabulka PolozkaObjPrij

Tato relace řeší vztah 1:1 mezi tabulkami *ObjPrij* a *ObjednavaneZbozi* a umožňuje tak vést v databázi podrobné data pro konkrétní druh zboží na základě objednávky. Primární klíč tvoří

sloupec *CisloPolozkyObjPrij*, a aby se zajistila jedinečnost, bude mít tvar číslo\_faktury/číslo\_položky, protože například číslo položky 1, se bude s největší pravděpodobností vyskytovat ve všech objednávkách.

Název atributu	Datový typ
CisloPolozkyObjPrij (PK)	Varchar(20)
KodZboziObj (FK)	Varchar(20)
CisloObjPrij (FK)	Varchar(20)
PocetKsObj	Numeric(4,1)

Obr 11: Tabulka PolozkaObjPrij

### Tabulka ObjednavaneZbozi

Tabulka *ObjednavaneZbozi* bude podávat podrobnější informace o objednávaném zboží. Přestože v danou chvíli, nebude určitý druh zboží objednán, údaje o tomto zboží v tabulce budou. Počet kusů daného zboží tedy může být 0. Primárním klíčem tabulky je *KodZboziObj*. Kód zboží představuje katalogové označení zboží u koncové dodavatelské firmy. Firma tento kód převezme a identifikuje jím přijaté zboží. Jedna položka tabulky, nebo-li jeden druh zboží, představuje součet objednaných kusů ze všech objednávek od různých odběratelů, avšak pro jednoho dodavatele. Protože tabulka je rodičovskou tabulkou atributu *KodZboziObj*, který je cizím klíčem v tabulce *PolozkaObjPrij*, jsou po přijetí např. objednávky od nového odběratele, nejprve zaevidovány údaje o odběrateli, údaje o objednavce v tabulce *ObjPrijata* a poté objednávané zboží, o kterém údaje v tabulce zůstanou i po vyřízení objednávky, jen počet ks bude 0. Pak se zavedou údaje do tabulky *Položka objednávky přijaté*.

Název atributu	Datový typ
KodZboziObj (PK)	Varchar(20)
ModelZboziObj	Varchar(20)
PopisZboziObj	Varchar(50)
CenaBezDPHks	Money
CenaBezDPHcelkem	Money
DPHobj	Int
DodIC (FK)	Char(8)
PocetZboziObj	Numeric(4,1)

Obr 12: Tabulka ObjednavaneZbozi

### Tabulka ObjednaneZboziOdberatelem

Do této tabulky jsou převedena některá data z tabulky *ObjednavaneZbozi* a to ty, na které je vytvořena a odeslána objednávka cílovému dodavateli. Převod dat se projeví v původní tabulce tak, že počet daného zboží se sníží o počet ks, které bylo objednáno. Primárním klíčem v tabulce je identifikační číslo.

Název atributu	Datový typ
KodZboziObj (FK)	Varchar(20)
PocetZboziObj	Numeric(4,1)
DodIC (FK)	Char(8)
IDzbozi (PK)	Int

Obr 13: Tabulka ObjednaneZboziOdberatelem

### Tabulka ObjZaslana

Data v této tabulce vychází z tabulky *ObjednaneZboziOdberatelem*. Tabulka obsahuje pouze tři sloupce, přičemž identifikátorem tabulky je číslo objednávky, které je ve tvaru měsíc\_rok/pořadové\_číslo\_objednávky\_v\_měsíci.

Název atributu	Datový typ
CisloObjZasl (PK)	Char(8)
DatumObjZasl	Date
DodIC (FK)	Char(8)

Obr 14: Tabulka ObjZaslana

### Tabulka PolozkaObjZasl

Tabulka na základě atributu *KodZboziObj* (cizí klíč) přináší podrobnější informace o objednávaném zboží jako model zboží, popis zboží, počet kusů. Primární klíč tvoří sloupec *CisloPolozkyObjZasl*, a aby se zajistila jedinečnost, bude mít tvar číslo\_faktury/číslo\_položky.

Název atributu	Datový typ
CisloPolozkyObjZasl (PK)	Char(11)
KodZboziObj (FK)	Varchar(20)
CisloObjZasl (FK)	Char(8)
PocetKsObj	Numeric(4,1)

Obr 15: Tabulka PolozkaObjZasl

### Tabulka Dodavatele

Tato tabulka bude obsahovat data o dodavatelích, se kterými firma pracuje nebo pracovala a nepřerušili úplně spolupráci. Primárním klíčem tabulky je *DodIC*. Tato identifikace byla jednoznačná, protože v tabulce se další jedinečné označení nevyskytuje. Aktuální počet dodavatelů je asi padesát. Primární klíč *DodIC* se jako cizí klíč vyskytuje ve třech jiných tabulkách. Tabulka je tedy v povinném vztahu se třemi dalšími relacemi.

Název atributu	Datový typ
DodNazev	Varchar(30)
DodIC (PK)	Char(8)
DodUlice	Varchar(20)
DodCP	Varchar(10)
DodMesto	Varchar(30)
DodPSC	Varchar(10)
DodStat	Varchar(20)
DodTelefon	Varchar(15)
DodPoznamka	Varchar(50)

Obr 16: Tabulka Dodavatele

### Tabulka FAP

Relace obsahuje data o přijatém zboží. Informuje o čísle faktury, datu přijetí, datu splatnosti a dodavateli.

Název atributu	Datový typ
CisloFAP (PK)	Varchar (20)
DatumPrijFAP	Date
DatumsplatFAP	Date
DodIC (FK)	Char(8)

Obr 17: Tabulka FAP

### Tabulka PolozkaFAP

V této tabulce je primárním klíčem atribut *CisloPolozkyFAP*. To od sebe odlišuje jednotlivé druhy zboží na faktuře. V tabulce jsou dále informace o počtu zboží a číslu faktury. Číslo položky je jako v předchozích případech ve tvaru *číslo\_faktury/číslo\_položky*.

Název atributu	Datový typ
CisloPolozkyFAP (PK)	Varchar(20)
CisloFAP (FK)	Varchar(20)
KodZboziPrij (FK)	Varchar(20)
PocetKsPrij	Numeric(4,1)

Obr 18: Tabulka PolozkaFAP

### Tabulka DodListPrij

Název atributu	Datový typ
CisloDodPrij (PK)	Varchar(20)
DatumPrij	Date
DodIC (FK)	Char(8)

Obr 19: Tabulka DodListPrij

### Tabulka PolozkaDodPrij

Tabulka prostřednictvím cizího klíče podává informace o dodaném zboží.

Název atributu	Datový typ
CisloPolozkyDodPrij (PK)	Varchar(20)
KodZboziPrij (FK)	Varchar(20)
CisloDodPrij (FK)	Varchar(20)
PocetKsPrij	Numeric(4,1)

Obr 20: Tabulka PolozkaDodPrij

### Tabulka PrijateZbozi

V tabulce jsou vedeny podrobné údaje o přijatém zboží – zboží na skladě. Podává informace jako je model zboží, popis zboží, počet kusů, ceny s daní a bez daně, váha zboží. Relace je prostřednictvím atributu *KodZboziPrij* ve vztahu s tabulkami *PolozkaDodListPrij*, *PolozkaFAP*, *PolozkaFAV* a *PolozkaDodListZasl*.

Název atributu	Datový typ
KodZboziPrij (PK)	Varchar(20)
ModelZboziPrij	Varchar(20)
PopisZboziPrij	Varchar(20)
PocetKsPrij	Numeric(4,1)
CenaBezDPHks	Money

CenaBezDPHcelkem	Money
DPHprij	Int
VahaKs	Numeric(4,2)
VahaCelkem	Numeric(4,2)
DodIC	Char(8)

Obr 21: Tabulka PrijateZbozi

### Tabulka FAV

Relace obsahuje data týkající se faktury, kterou vystavuje firma pro odběratele. Primárním klíčem je číslo faktury, které je ve tvaru měsíc\_rok/*F*\_pořadové\_číslo\_faktury.

Název atributu	Datový typ
CisloFAV (PK)	Char(9)
DatumZasl	Date
DatumSplat	Date
OdbIC (FK)	Char(8)

Obr 22: Tabulka FAV

### Tabulka PolozkaFAV

Název atributu	Datový typ
CisloPolozkyFAV (PK)	Char(11)
KodZbozPrij (FK)	Varchar(20)
CisloFAV (FK)	Char(8)
PocetZboziPrij	Numeric(4,1)

Obr 23: Tabulka PolozkaFAV

### Tabulka DodListZasl

Spolu s fakturou vystavuje firma pro odběratele dodací list. Tabulka je ve vztahu s tabulkou *PolozkaDodZasl* a na základě cizího klíče *KodZboziPrij* vyjadřuje informace o zboží, jako je např. váha zboží.

Název atributu	Datový typ
CisloDodZasl (PK)	Varchar(8)
DatumZasl	Date
OdbIC (FK)	Char(8)

Obr 24: Tabulka DodListZasl

### Tabulka PolozkaDodZasl

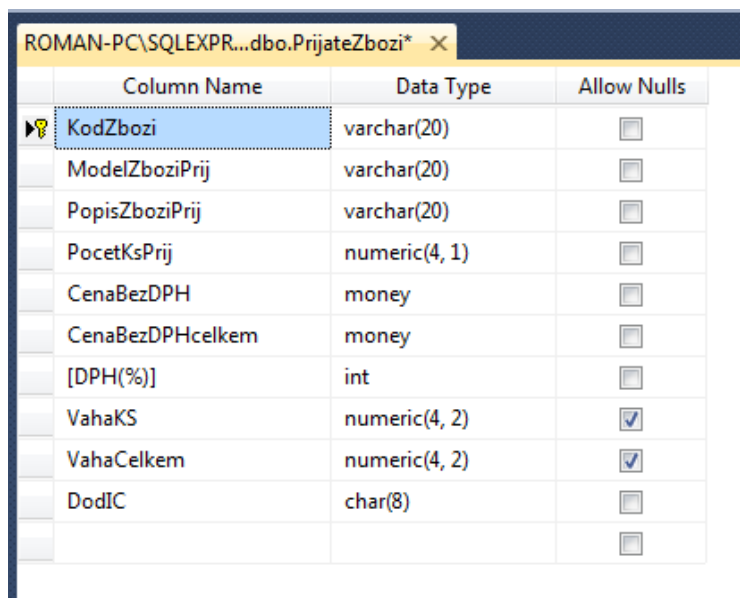
Název atributu	Datový typ
CisloPolozkyDodZasl (PK)	Char(11)
KodZboziPrij (FK)	Varchar(20)
CisloDodZasl (FK)	Varchar(20)
PocetKsZasl	Numeric(4,1)

Obr 25: Tabulka PolozkaDodZasl



## 4.4 Fyzický návrh databáze

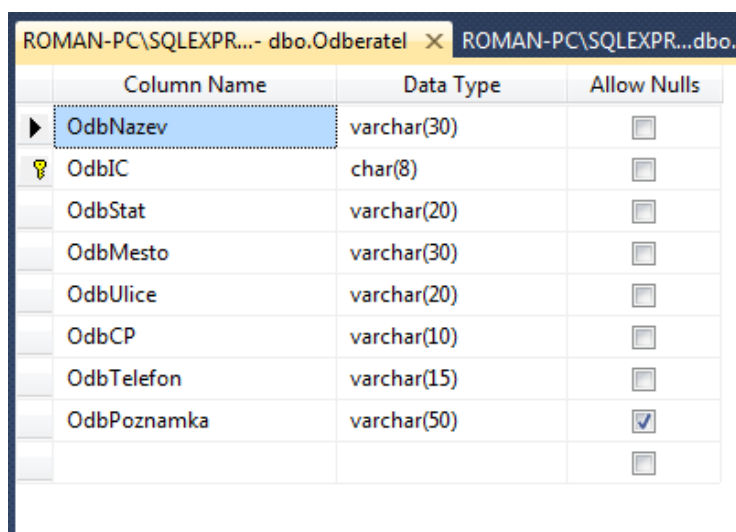
Relační databáze byla vytvořena v SQL serveru. Na následujících obrázcích je ukázka třech relací: *Přijaté zboží*, *Odběratelé*, *Položka na dodacím listě přijatém*.



The screenshot shows the 'PrijateZbozi' table in the design view. The table has the following columns and properties:

Column Name	Data Type	Allow Nulls
KodZbozi	varchar(20)	<input type="checkbox"/>
ModelZboziPrij	varchar(20)	<input type="checkbox"/>
PopisZboziPrij	varchar(20)	<input type="checkbox"/>
PocetKsPrij	numeric(4, 1)	<input type="checkbox"/>
CenaBezDPH	money	<input type="checkbox"/>
CenaBezDPHcelkem	money	<input type="checkbox"/>
[DPH(%)]	int	<input type="checkbox"/>
VahaKS	numeric(4, 2)	<input checked="" type="checkbox"/>
VahaCelkem	numeric(4, 2)	<input checked="" type="checkbox"/>
DodIC	char(8)	<input type="checkbox"/>

Obr 26: Tabulka Přijaté zboží v návrhovém zobrazení



The screenshot shows the 'Odberatel' table in the design view. The table has the following columns and properties:

Column Name	Data Type	Allow Nulls
OdbNazev	varchar(30)	<input type="checkbox"/>
OdbIC	char(8)	<input type="checkbox"/>
OdbStat	varchar(20)	<input type="checkbox"/>
OdbMesto	varchar(30)	<input type="checkbox"/>
OdbUlice	varchar(20)	<input type="checkbox"/>
OdbCP	varchar(10)	<input type="checkbox"/>
OdbTelefon	varchar(15)	<input type="checkbox"/>
OdbPoznamka	varchar(50)	<input checked="" type="checkbox"/>

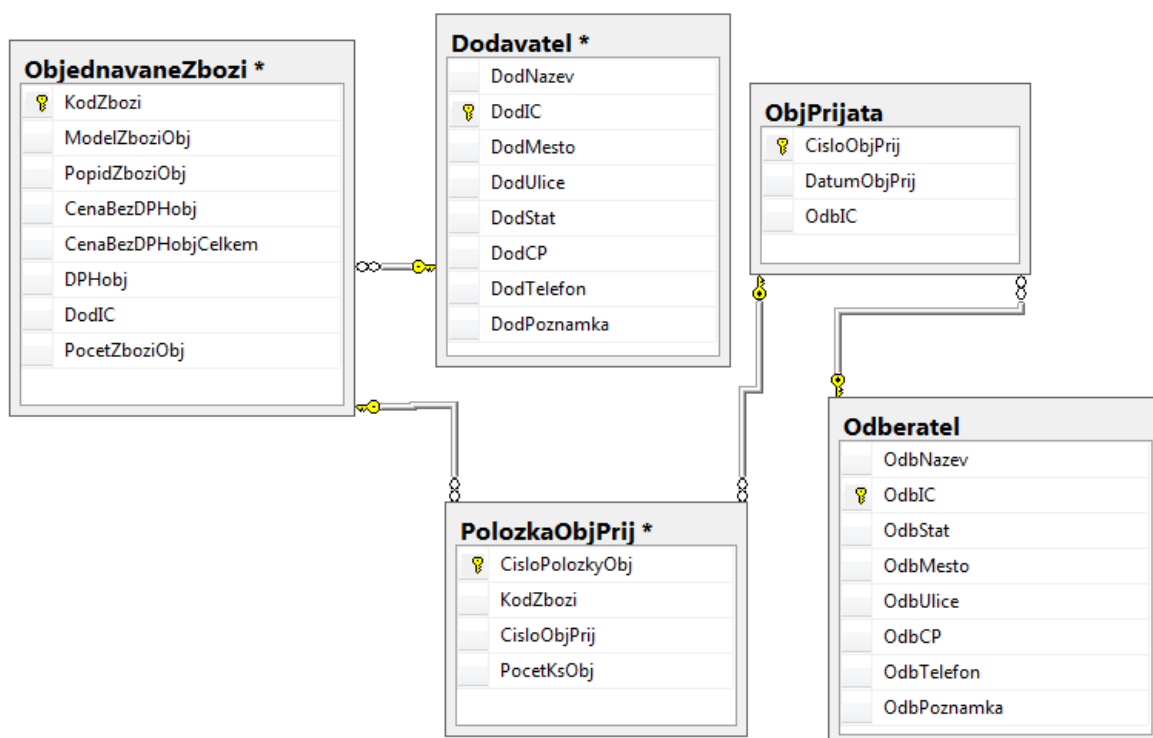
Obr 27: Tabulka Odběratel v návrhovém zobrazení

ROMAN-PC\SQLXP...o.PolozkaDodPrij			
	Column Name	Data Type	Allow Nulls
	CisloPolozkyDodPrij	varchar(20)	<input type="checkbox"/>
	KodZbozi	varchar(20)	<input type="checkbox"/>
	CisloDodPrij	varchar(20)	<input type="checkbox"/>
	PocetKsPrij	numeric(4, 1)	<input type="checkbox"/>

Obr 28: Tabulka Položka na dodacím listě přijatém v návrhovém zobrazení

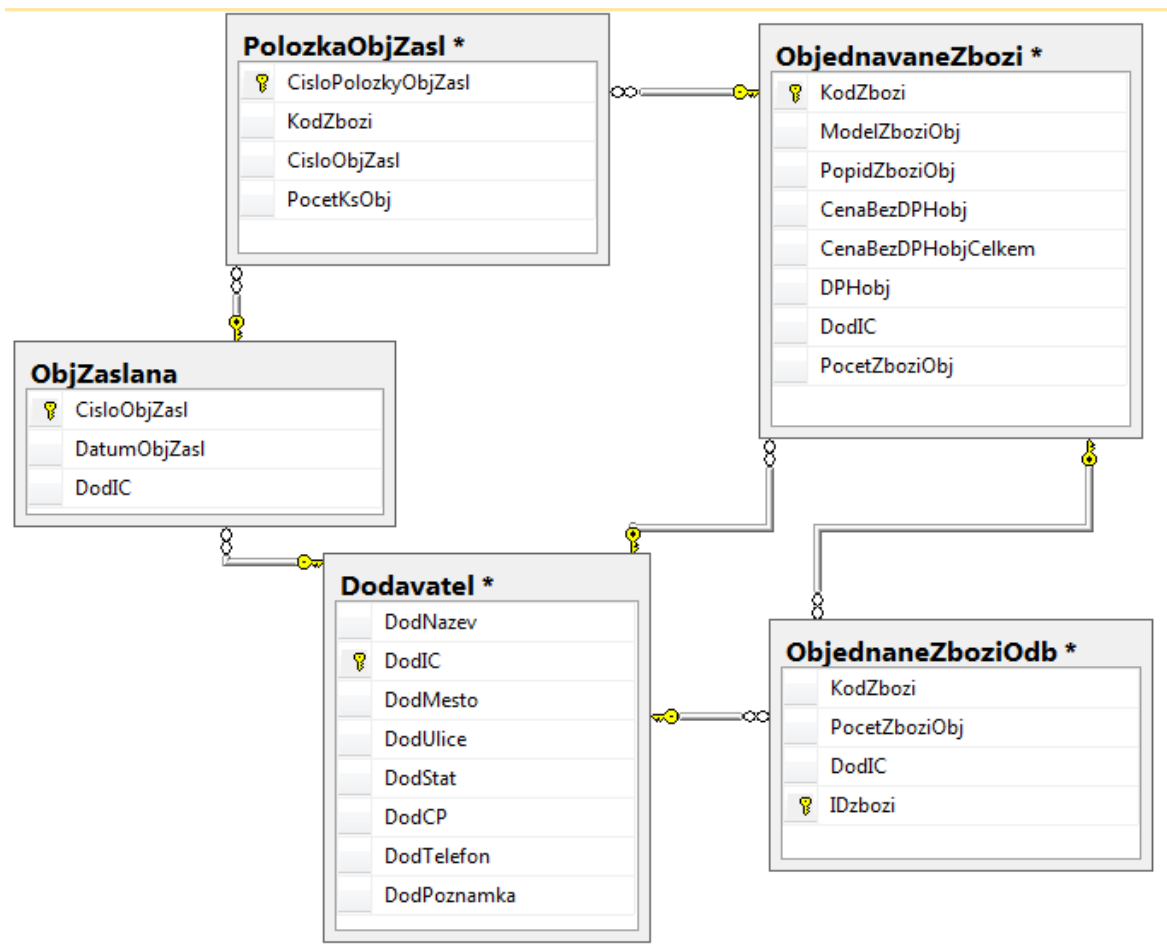
#### 4.4.1 Vztahy mezi tabulkami

Následující obrázek ukazuje vztahy mezi tabulkami, týkající se procesu přijetí objednávky a zaevidování údajů z objednávky – tedy informace o odběrateli, o dodavateli, údaje o objednaném zboží, objednávce samotné a o konkrétních položkách.



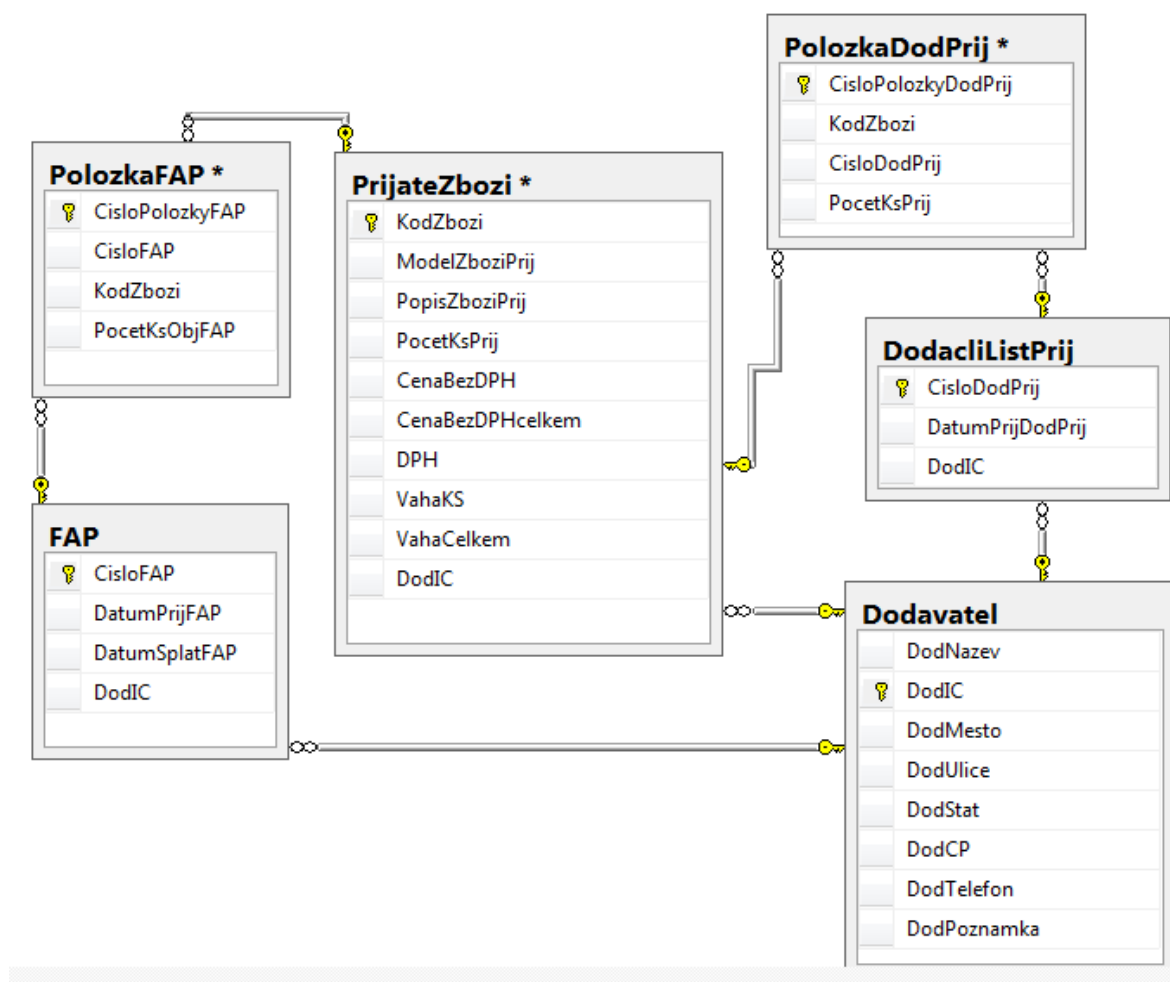
Obr 29: Propojení tabulek

Na obrázku (obrázek 27) je vidět propojení tabulek v procesu objednání zboží od koncového dodavatele.



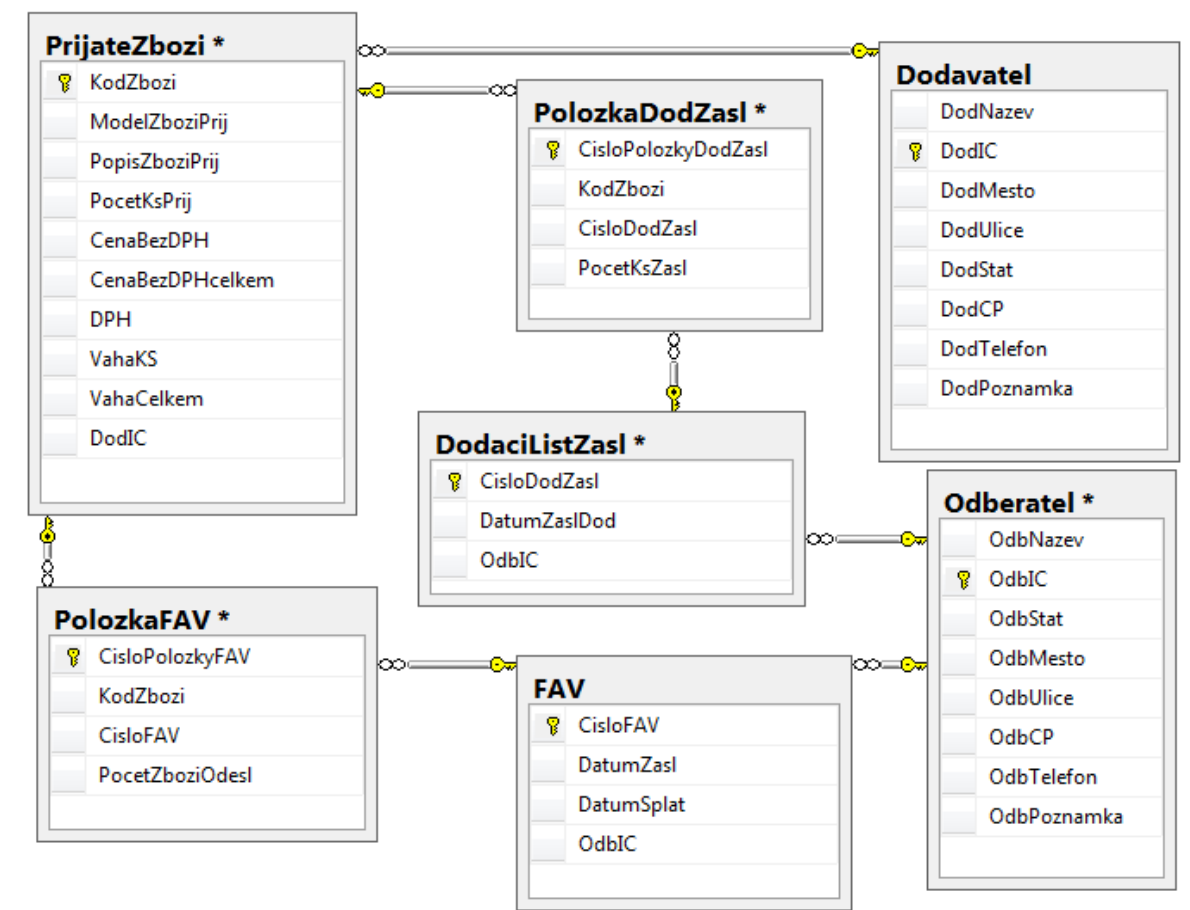
Obr 30: Propojení tabulek

Proces přijetí zboží do firmy spolu s fakturou a dodacím listě je vyobrazen pomocí tabulek a jejich propojení na obrázku pod textem (obrázek 28). Po přijetí dodávky se zaevidují informace o přijatém zboží, údaje o přijatém dodacím listě, faktuře a jejich konkrétních položkách.



Obr 31: Propojení tabulek

Posledním procesem je odeslání dodávky odběrateli. Po přijetí kamionu se dle objednávek nachystané várky naloží do kamionu spolu s fakturou a dodacím listem vydaným firmou. Údaje o zboží na skladě (přijatém zboží), o faktuře, dodacím listě a jejich položkách jsou evidovány v tabulkách na následujícím obrázku. Celý databázový diagram je v příloze č. 1.



Obr 32: Propojení tabulek

### 4.4.2 Uložené procedury

Pro efektivnější a rychlejší práci s daty byly v databázi vytvořeny uložené procedury. Například uložená procedura s názvem *NovyDodavatel* má za úkol urychlit zadávání nových dodavatelů do databáze. Ukázka kódu vytvořené uložené procedury je na následujícím obrázku.

```
CREATE PROC NovyDodavatel (  
    @DodNazev varchar(30),  
    @DodIC char(8),  
    @DodMesto varchar(30),  
    @DodUlice varchar(20),  
    @DodStat varchar(20),  
    @DodCP varchar(10),  
    @DodTelefon varchar(15),  
    @DodPoznamka varchar(50))  
AS  
BEGIN  
    IF NOT EXISTS (SELECT * FROM Dodavatel WHERE DodIC=@DodIC)  
    BEGIN  
        INSERT INTO [dbo].[Dodavatel] (DodNazev, DodIC, DodMesto, DodUlice, DodStat, DodCP, DodTelefon, DodPoznamka)  
        VALUES (@DodNazev, @DodIC, @DodMesto, @DodUlice, @DodStat, @DodCP, @DodTelefon, @DodPoznamka)  
    END  
END
```

Obr 33: Ukázka kódu tvorby uložené procedury *NovyDodavatel*

## 5 Závěr

Cílem bakalářské práce bylo provést analýzu firmy zabývající se logistikou, navrhnout pro ni databázový systém, který by řešil situaci, ve které je dosavadní řešení nedostačující a navrhnout systém implementovat.

Nejprve bylo na problematiku návrhu databáze pohlíženo teoreticky. Obecně byl popsán koncept databáze a další pojmy s tím související. Byl proveden teoretický popis metodiky návrhu databáze a jejích dílčích částí a v neposlední řadě také popis provádění datové analýzy a jazyka SQL, který byl použit při implementaci databáze.

V druhé části práce byl cílem samotný návrh budoucího systému. Na základě rozhovoru vedeného se zaměstnanci firmy, požadavků jimi definovaných a prostudováním firemních dokumentů jsem provedla analýzu stávající situace pomocí analytického nástroje Data Flow Diagram. Na základě definovaných datových toků, datových skladů a procesů znázorněných v diagramu vzešly informace o tom, jaké data a v jaké souvislosti by měla firma uchovávat a zpracovávat.

Na bázi informací zjištěných v analýze byl pak v konceptuálním návrhu vytvořen entitně-relační diagram s primárními klíči a atributy a zaveden slovník dat. V druhém kroku, tedy v logickém návrhu databáze pak byl vytvořen ER diagram obsahující však už i cizí klíče. V druhé části logického návrhu byly na základě informací vycházejících ze slovníku dat a ER diagramu navrženy a vytvořeny struktury tabulek, které jsou podkladem v procesu samotné implementace databáze.

Ve třetí fázi návrhu databáze pak již byly pomocí SQL serveru vytvořeny tabulky a vztahy mezi nimi. V závěru této podkapitoly byla také zmíněna tvorba procedur v databázi, které mají ulehčit a především zrychlit práci s daty.

Vzhledem k výše uvedenému bylo cíle bakalářské práce dosaženo.

Do budoucna se počítá s rozvojem databázového systému. Vytvořená databáze bude dále sloužit jako podklad pro aplikaci, která databázi bude používat. Aplikace bude vytvořena v jazyce Java IT specialistou, který doplní další potřebné funkce a vytvoří uživatelské rozhraní.

## Seznam použité literatury

### Odborná kniha

CONOLLY, T., C. E. BEGG a R.HOLOWCZAK. *Mistrovství - databáze: profesionální průvodce tvorbou efektivních databází*. Brno: Computer Press, 2009. ISBN 978-80-251-2328-7.

HERNANDEZ, Michael. J. *Návrh databáz: základy databází: krok za krokem*. Praha: Grada, 2006. ISBN 80-247-0900-7.

HOTEK, Mike. *Microsoft SQL Server 2008: krok za krokem*. Brno: Computer Press, 2009. ISBN 978-80-251-2466-6.

SHELDON, Robert. *SQL začínáme programovat*. Praha: Grada 2005. ISBN 80-247-0999-6

ŠIMONOVÁ Stanislava, P. J. *Databázové systémy I pro kombinovanou formu studia*. Pardubice: Univerzita Pardubice 2007

### Elektronické dokumenty

BI EXPERTS. *Normalizace*. [5. 4. 2013]. Dostupné z:

<http://www.biexperts.cz/index.php/cs/professional/18-ctsql/31-arnormalization.html>

DATA FLOW DIAGRAM. [19. 4. 2013]. Dostupné z:

[http://is.muni.cz/th/4487/fi\\_m/html/ch04s03.html](http://is.muni.cz/th/4487/fi_m/html/ch04s03.html)

DATABÁZOVÉ SYSTÉMY. *Relační databáze a ER diagramy*, [5. 4. 2013]. Dostupné z:

<http://www.toker.cz/pages/diagramy.html>

INFORMAČNÍ SYSTÉMY. *Kardinalita vztahu*. [20. 3. 2013]. Dostupné z:

<http://informacni-technologie.studentske.cz/2009/02/kardinalita-vztahu.html>

INFORMAČNÍ SYSTÉMY. *Sémantický datový model*. [19. 4. 2013]. Dostupné z:

<http://informacni-systemy.studentske.cz/2009/07/1-semanticky-datovy-model.html>

INTERVAL. *Databáze a jazyk SQL*. [19. 4. 2013]. Dostupné z:

<http://interval.cz/clanky/database-a-jazyk-sql/>



KONTEXTOVÝ DIAGRAM. [5. 4. 2013]. Dostupné z:

<http://zdenek2.euweb.cz/doc3/prois32a.html>

SKŘIVÁNEK, F. (23. 10 2008). *Databázový svět*. [15. 4. 2013] Získáno z

<http://www.dbsvet.cz/view.php?cisloclanku=2008102302>

ZENDULKA, J. *Databázové systémy a návrh databází – 2 Konceptuální modelování a návrh databáze*. [19. 4. 2013]. Dostupné

z:[http://www.fit.vutbr.cz/study/courses/DSI/public/pdf/nove/2\\_kmod.pdf](http://www.fit.vutbr.cz/study/courses/DSI/public/pdf/nove/2_kmod.pdf)

## **Seznam zkratek**

1NF – první normální forma

2NF – druhá normální forma

3NF – třetí normální forma

DBMS – systém řízení báze dat

DFD – Data Flow Diagram

DPH – daň z přidané hodnoty

ERM – entity-relationship model

IČ – identifikační číslo

IS – informační systém

IT – informačních technologie

KS - kus

MS – Microsoft

NF – normální forma

RDBMS – relační systém řízení báze dat

SQL – structuredquerylanguage

SŘBD – systém řízení báze dat

## Prohlášení o využití výsledků mé bakalářské práce

Prohlašuji, že

- jsem byl(a) seznámen(a) s tím, že na mou diplomovou (bakalářskou) práci se plně vztahuje zákon č. 121/2000 Sb. – autorský zákon, zejména § 35 – užití díla v rámci občanských a náboženských obřadů, v rámci školních představení a užití díla školního a § 60 – školní dílo;
- beru na vědomí, že Vysoká škola báňská – Technická univerzita Ostrava (dále jen VŠB-TUO) má právo nevýdělečně, ke své vnitřní potřebě, diplomovou (bakalářskou) práci užít (§ 35 odst. 3);
- souhlasím s tím, že diplomová (bakalářská) práce bude v elektronické podobě archivována v Ústřední knihovně VŠB-TUO a jeden výtisk bude uložen u vedoucího diplomové (bakalářské) práce. Souhlasím s tím, že bibliografické údaje o diplomové (bakalářské) práci budou zveřejněny v informačním systému VŠB-TUO;
- bylo sjednáno, že s VŠB-TUO, v případě zájmu z její strany, uzavřu licenční smlouvu s oprávněním užít dílo v rozsahu § 12 odst. 4 autorského zákona;
- bylo sjednáno, že užít své dílo, diplomovou (bakalářskou) práci, nebo poskytnout licenci k jejímu využití mohu jen se souhlasem VŠB-TUO, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly VŠB-TUO na vytvoření díla vynaloženy (až do jejich skutečné výše).

V Ostravě dne 10. 5. 13 .....

.....  
Tereza Pavlová

jméno a příjmení studenta

## **Seznam příloh**

Příloha 1: Slovník dat

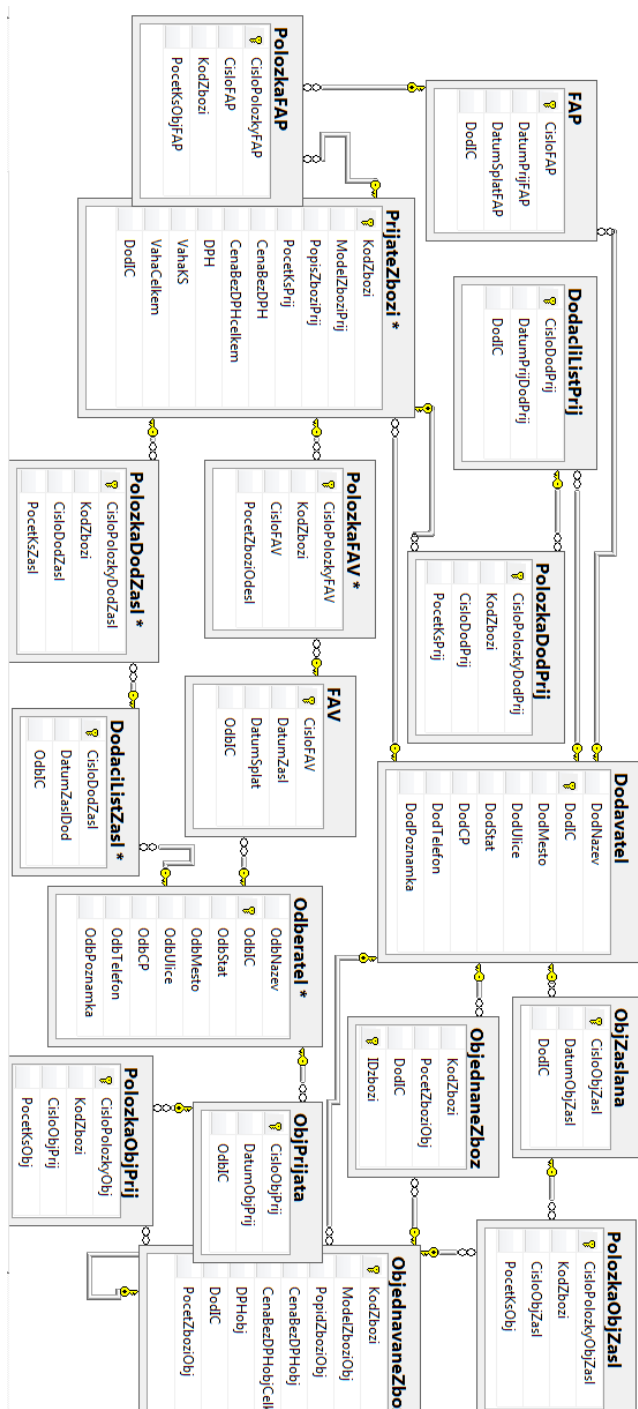
Příloha 2: Databázový diagram

Příloha 1: Slovník dat

Název tabulky	Název sloupce	Popis sloupce	Datový typ	Null
<b>FAP</b>	CisloFAP	Číslo faktury přijaté	Varchar, 20	Ne
	DatumPrij	Datum přijetí faktury	Date	Ano
	DatumSplat	Datum splatnosti faktury	Date	Ne
<b>DodaciListPrij</b>	CisloDodPrij	Číslo dodacího listu přijatého od dod.	Varchar, 20	Ne
	VahaDodPrij	Váha dodaného zboží	Numeric, 4,2	Ne
<b>PolozkaFaP</b>	CisloPolozkyFAP	Číslo položky na faktuře přijaté od dod.	Varchar, 20	Ne
	CisloFAP	Číslo faktury přijaté	Varchar, 20	Ne
	KodZboziPrij	Kód přijatého zboží	Varchar, 20	Ne
	PocetKsPrij	Počet ks přijatého zboží	Numeric, 4,1	Ne
<b>PolozkaDodPrij</b>	CisloPolozkyDodPrij	Číslo položky na dodacím listě přijatém	Integer	Ne
	KodZboziPrij	Kód přijatého zboží	Varchar, 20	Ne
	CisloDodListPrij	Číslo dodacího listu přijatého	Varchar, 20	Ne
	PocetKsPrij	Počet ks přijatého zboží	Numeric, 4,2	
<b>PrijateZbozi</b>	KodZboziPrij	Kód přijatého zboží	Varchar, 20	Ne
	ModelZboziPrij	Model přijatého zboží dle dodacího listu	Varchar, 20	Ne
	PocetKsPrij	Počet ks přijatého zboží	Numeric, 4,1	Ne
	PopisZboziPrij	Popis přijatého zboží dle dodacího listu	Varchar, 20	Ne
	CenaBezDPH	Cena za kus zboží bez DPH dle faktury přij.	Money	Ne
	CenaBezDPHcelkem	Cena celková za zboží bez DPH dle faktury	Money	Ne
	DPHprij	Daň z přidané hodnoty přijatého zboží	Integer	Ne
	VahaKs	Váha zboží za jeden ks	Numeric, 4,2	Ne
	VahaCelkem	Váha celková za druh	Numeric, 4,2	Ne

		zboží		
	OdbIC	IČ odběratele	Char, 8	Ne
<b>PolozkaFaV</b>	CisloPolozkaFaV	Číslo položky na faktuře vydané	Integer	Ne
	PocetZboziOdesl	Počet ks odesílaného zboží	Numeric, 4,1	Ne
<b>PolozkaDodZasl</b>	CisloPolozkyDodZasl	Číslo položky na dodacím listě zasl.	Char, 11	Ne
	PocetKsZasl	Počet ks odesílaného zboží	Numeric, 4,1	Ne
<b>FAV</b>	CisloFAV	Číslo faktury vystavené	Char, 8	Ne
	DatumZasl	Datum vystavení faktury	Date	Ne
	DatumSplat	Datum splatnosti faktury	Date	Ne
<b>DodaciListZasl</b>	CisloDodZasl	Číslo dodacího listu vystaveného	Char, 8	Ne
	DatumZasl	Datum vystavení dodacího listu	Date	Ne

## Príloha 2: Databázový diagram



Obr 34: Databázový diagram